

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO DE FIN DE GRADO

Integración de un ECM con el sistema de ventas de Amazon Marketplace.

Javier Gutiérrez Navío

Tutor: Roberto López Santos

Ponente: Miren Idoia Alarcón Rodríguez

Julio 2016

**Integración de un ECM con el sistema de ventas de Amazon
Marketplace.**

1516_110_ISSI

**AUTOR: Javier Gutiérrez Navío
TUTOR: Roberto López Santos
PONENTE: Miren Idoia Alarcón Rodríguez**

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2016**

Resumen

Vivimos en una sociedad 3.0, donde internet cada vez va tomando una posición más importante en nuestras vidas. Cosas que antes se hacían, como comprar, ahora ya se hacen cada vez más por Internet.

Dentro del amplio mundo de las compras en internet, existe Amazon, el mayor portal de ventas online que existe. Ahí, los comerciantes pueden anunciar sus productos, y al ser un portal tan grande adquieren más visibilidad de la que podrían tener por sus propios medios.

El objetivo de este proyecto consiste en desarrollar una integración de una tienda online con Amazon, de manera que el comerciante pueda gestionar desde un único sitio los productos y pedidos de su tienda y los que tienen en Amazon, dándole una mayor posibilidad de ventas.

Esta integración se ha planteado como una mejora del producto de comercio electrónico de la empresa Yunbit, desarrollándose inicialmente para un solo cliente, Alma en pena, una tienda internacional de zapatos, pero planteándose como un módulo genérico que se pueda vender a futuros clientes.

Este módulo ofrece al cliente la creación de listados de objetos para su publicación de Amazon, gestión de sus características, precios y presencia dentro de la tienda online. Además, integra servicios para automatizar la sincronización con Amazon y la gestión de los pedidos del mismo, haciéndose transparentes para el cliente. Para el desarrollo de este proyecto, se ha partido del proyecto de tienda online que la empresa Yunbit había desarrollado para Alma en pena, y mediante la creación de clases de objetos, gestores, tareas automáticas, y la modificación de dicho proyecto se ha elaborado este módulo. El proyecto ha sido enteramente hecho en PHP, y se han usado bases de datos de MySQL, mensajes en XML y validación con XSD. Para la comunicación entre servidores se ha usado un sistema rest mediante mensajes enviados con CURL. Para la prevención de DDoS, Amazon usa un sistema de throttling (un sistema de regulación para la recepción de paquetes), y se ha elaborado una prevención en caso de que se llegue al límite de mensajes admitidos por este sistema, retrasando los envíos de mensajes de manera exponencial, como si fuera un goteo, para que el sistema pueda recuperar su capacidad.

Este proyecto ha sido validado por el cliente, con unos resultados muy positivos, y por los jefes de la empresa Yunbit, y se ha propuesto una futura ampliación y una venta a la bolsa de clientes de la empresa, al ser una mejora muy potente del sistema de gestión de comercios electrónicos de Yunbit.

Palabras clave

Amazon, Venta online, Ecommerce, Marketplace, Back-end, Front-end, Mercader, Framework, GeaWebBlocks, Businesscloud, Alma en pena, Listing, Mock, MVC, Stock, Throttling.

Abstract

We live in a society where internet is becoming more and more important in our lives. Things we used to do in person, like buying, now they are become increasingly more so online, more comfortably without leaving home.

Within the wide world of shopping online, there is Amazon, the largest online sales portal that exists. There, merchants can advertise their products, and being such a large site gain more visibility than they could have on their own, that means more orders.

The objective of this project is to develop an integration of an online store with Amazon, so that the trader can manage from one site products and orders from their store and those in Amazon, giving it a greater chance of sales.

This integration has been raised as improvement of the e-commerce of the company Yunbit, developed initially for a single customer, Alma en pena, an international shoe store, but considering as a generic module that can be sold to other clients.

This module offers customers the creation of lists of objects for publication of Amazon, management characteristics, their prices and presence within the online store. It also integrates services to automate synchronization with amazon and order management thereof, becoming transparent to the client.

For the development of this project, we part from the project online store that the company Yunbit had developed for Alma en pena, and by creating object classes, managers, automated tasks, and modifying the project has been, it's been created this module. The project has been entirely made in PHP, and has been used MySQL databases, messages in XML and XSD validation. For communication between servers it's been used a rest system messages sent by CURL. To prevent DDOS, Amazon uses throttling (a control system for receiving packets) and has developed prevention if the limit is reached supported by this system messages, delaying shipments posts exponentially, like a trickle, so that the system can recover its capacity.

This project has been validated by the customer, with very positive results, and the heads of the company Yunbit, and has proposed future expansion and sale to the stock company clients, being a very powerful enhancement electronic management system stores Yunbit.

Amazon, Venta online, Ecommerce, Marketplace, Back-end, Bulletpoint, Mercader, Framework, GeaWebBlocks, Businesscloud, Alma en pena, Front-end, Listing, Mock, MVC, Stock, Throttling.

Keywords

Amazon, Online marketing, Ecommerce, Marketplace, Back-end, Front-end, Merchant, Framework, GeaWebBlocks, Businesscloud, Alma en pena, Listing, Mock, MVC, Stock, Throttling

Agradecimientos

Para empezar, me gustaría agradecer a todos los profesores de la EPS que me han enseñado en qué consiste el ser un ingeniero informático, a pesar de lo duro del proceso, ya que sin ellos no habría podido llegar a donde estoy ahora mismo.

De segundo, quiero agradecer a todos mis compañeros de fatigas durante los 4 años, especialmente a Diego Guerra, mi compañero de prácticas y de oficinas durante casi dos años, que me ha apoyado y ha compartido mis insomnios por las benditas prácticas.

También me gustaría agradecer a mi hermano gemelo, Daniel Gutiérrez, con el que he cursado la carrera y que hemos compartido andanzas, aunque la vida nos haya llevado al final por distintos senderos.

Cabe destacar a la empresa Yunbit, en la que entré como becario, y en la que llevo trabajando tres años, que me ha dado la oportunidad de crecer profesionalmente y adquirir grandes responsabilidades, a pesar de mi corta edad. Dentro de ella, cabe mencionar a los jefes, Roberto López y Juan Conde, y al director tecnológico, Mario Villar. También a mis compañeros Jaime Acevedo, Daniel González, Jesús Vázquez, Marta Núñez y José Miguel Gil, sin los que no hubiera llegado a este punto.

Por último, me gustaría agradecer a una persona que se ha ofrecido, aun a pesar de todo lo que tiene, a tutorizarme este TFG y me ha ayudado en todo el proceso, además de toda la inestimable ayuda que ha dado durante toda la carrera. Muchas gracias Miren Idoia.

“La vida no consiste en esperar a que pase la tormenta, es aprender a bailar bajo la lluvia”

“Vive como si fueras a morir mañana. Aprende como si fueras a vivir para siempre”

INDICE DE CONTENIDOS

1	Introducción	1
1.1	Motivación.....	1
1.2	Visión Global.....	2
1.3	Propósito	2
1.4	Organización de la memoria	3
2	Estado del arte	5
2.1	Introducción	5
2.2	Tecnologías usadas	6
2.2.1	PHP.....	6
2.2.2	MySQL.....	7
2.2.3	GeaWebBlocks.....	8
2.3	Amazon Marketplace Web Service	9
2.4	Conclusión	10
3	Definición del proyecto y objetivos	13
3.1	Modelo de datos.....	14
3.1.1	Productos	14
3.1.2	Pedidos	15
3.1.3	Cliente	15
3.2	Gestores y modelo de datos	16
3.3	Tareas de sincronización.....	16
3.3.1	Sincronización de productos	16
3.3.2	Carga manual.....	16
3.4	Reportes y alertas.....	17
4	Análisis y Diseño	19
4.1	Análisis	19
4.1.1	Pedidos	19
4.1.2	Productos	20
4.1.3	Roles de usuarios.....	21
4.1.4	Requisitos funcionales.....	21
4.1.5	Requisitos no funcionales.....	22
4.2	Diseño	23
4.2.1	Base de datos.	25
4.2.2	Gestores de datos.....	26
4.2.3	Modelado de productos.	26
4.2.4	Sincronización de productos.	27
4.2.5	Sincronización de pedidos.....	27
4.2.6	Alertas	28
4.2.7	Precios	28
5	Implementación	29
5.1	Productos	29
5.1.1	Modelo de datos de Amazon Marketplace	29
5.1.2	Modificaciones en los productos	31
5.2	Sincronización	32
5.2.1	Modalía.....	32

5.2.2	Amazon MWS.....	32
5.3	Pedidos.....	33
5.3.1	Pedidos de Amazon.....	33
5.3.2	Actualización de stock	34
5.4	Interfaz del usuario	34
5.4.1	Gestores de Listings	35
5.4.2	Gestores de productos	37
5.4.3	Gestores de pedidos.....	38
5.5	Alerta de precios	39
6	Pruebas y validación	41
6.1	Estrategia de pruebas	41
6.2	Desarrollo de las pruebas	41
6.2.1	Pruebas unitarias	41
6.2.2	Pruebas de integración	42
6.2.3	Pruebas de usabilidad	42
6.3	Validación.....	43
7	Conclusiones y trabajo futuro.....	45
7.1	Conclusiones.....	45
7.2	Trabajo futuro	45
8	Glosario	47
9	Referencias.....	49
10	Anexos	I
A	Ejemplo de comunicación de Productos con Amazon.....	I
A.1	Mensaje de producto.....	I
A.2	Mensaje de imágenes	II
A.3	Mensaje de inventario (Stock)	II
A.4	Mensaje de relaciones.....	III
B	Ejemplo de comunicación de Pedidos con Amazon.....	V
B.1	Pedido realizado.....	V
B.2	Productos de un pedido.....	VI
B.3	Aceptación de un pedido	VII
B.4	Rechazo de un pedido	VII
B.5	Cumplimentación de envío de un pedido	VIII
B.6	Cancelación de un producto dentro de un pedido.....	VIII
C	Listado de links para los XSDs.....	X
D	Ampliación del diagrama de clases para los Productos.....	XI

INDICE DE FIGURAS

FIGURA 1 : LÍNEA TEMPORAL DE PHP.....	6
FIGURA 2: LÍNEA TEMPORAL DE MYSQL.....	8
FIGURA 3: DIRECTORIOS DE WebBLOCKS.....	9
FIGURA 4 : CICLO DE VIDA DE UN PEDIDO	19
FIGURA 5: VIDA DE UN PEDIDO, DESDE QUE SE CREA HASTA QUE SE REALIZA EL COBRO	20
FIGURA 6: TAREAS EJERCIDAS POR CADA SERVIDOR.....	24
FIGURA 7: INTERACCIÓN ENTRE LOS DISTINTOS SERVIDORES	24
FIGURA 8: ESTRUCTURA DE LA DB	26
FIGURE 9: CICLO DE VIDA DE UN PEDIDO EN EL SISTEMA	28
FIGURA 10: DIAGRAMA DE LOS OBJETOS DESARROLLADOS POR EL ALUMNO.....	30
FIGURA 11: ÁRBOL DE CATEGORÍAS DE AMAZON CREADO POR EL ALUMNO	31
FIGURA 12: LISTADO DE LISTINGS	35
FIGURA 13: DATOS GENERALES DEL GESTOR DE LISTINGS.....	35
FIGURA 14: GESTOR DE STOCK	36
FIGURA 15: AGREGADOR DE PRODUCTOS	36
FIGURA 16: LISTADO DE LOGS DE SUBIDA.....	36
FIGURA 17: GESTOR DE CARACTERÍSTICAS Y BULLETPPOINTS DE UN PRODUCTO.....	37
FIGURA 18: LISTADO DE PEDIDOS	38
FIGURA 19: GESTOR DE PEDIDOS.....	38

1 Introducción

1.1 Motivación

Esta memoria abarca un proyecto cuyo propósito es la ampliación de un Ecommerce previamente desarrollado por la empresa Yunbit S.L. y su integración con el Marketplace Web Service de Amazon. Los objetivos del desarrollo correspondiente a este proyecto son:

- Permitir la sincronización con el proveedor de datos que nos proporciona periódicamente la información acerca de los productos de la empresa cliente, Alma en pena, a través de un servicio ofrecido por un tercero, Modalia,
- Permitir una centralización de los servicios de gestión de productos y pedidos a través del gestor de Yunbit.
- Implementar un servicio que sincronice los datos de los productos con el inventario de Amazon.
- Mantener sincronizado y actualizado este inventario de forma transparente para el cliente.
- Implementar un servicio que permita la automatización de la gestión de los envíos hechos a través de la plataforma de Amazon.
- Realizar este módulo de forma genérica para el producto de Yunbit (sin demasiada personalización, y personalizado mediante parámetros y herencia) para facilitar su desarrollo en el producto desarrollado para otros clientes.

Este proyecto surge como iniciativa propia del alumno, y es presentado a su jefe, Roberto López, de la empresa Yunbit, en la cual está trabajando en el momento del desarrollo del proyecto. El proyecto fue aprobado desde un principio, pero no se dio el visto bueno hasta poder vendérselo a un cliente. Este cliente ha sido Alma en pena, una empresa fabricante y distribuidora de zapatos de mujer a nivel europeo (<http://www.almaenpena.com>) al que previamente la empresa Yunbit había desarrollado su Ecommerce. Todos los datos presentes en este documento pertenecen a dicha empresa, salvo algún caso de prueba, y se le ha dado previo permiso al alumno para utilizar cualquier dato no confidencial, por lo que se omitirán únicamente claves privadas, contraseñas, datos bancarios y direcciones.

El sistema ha sido desarrollado en PHP, usando dos frameworks propios de Yunbit: *GeaWebBlocks*, un framework basado en *PHPbb* para el desarrollo de los gestores de datos y manejo de plantillas, y *BussinesCloud*, un sistema propio, también en PHP, que funciona como MVC, para la modelización de los datos.

Este proyecto ha supuesto una remuneración económica para el alumno y para la empresa, pero por motivos de confidencialidad y privacidad, se van a omitir estos datos, tales como el precio de venta del proyecto como los presupuestos del mismo.

Durante este desarrollo, realizado por el alumno, se ha contado tanto con el apoyo del equipo propio de Yunbit y del cliente, que han colaborado para la definición de los objetivos y requisitos del mismo, como con la documentación ofrecida por el servicio de soporte al desarrollador de Amazon, si bien este servicio ha resultado escaso y poco satisfactorio.

1.2 *Visión Global*

Para poder realizar el trabajo mencionado en el apartado anterior, y que se describirá durante la memoria, ha sido necesario, previamente, realizar un estudio de los modelos de datos y servicios de la API de Amazon. Tras ello, y en primer lugar, ha habido que desarrollar nuevas tablas y modificar el modelo de la base de datos para poder almacenar la información deseada, y, en segundo lugar, crear herramientas para el acceso y fácil edición de los datos por parte del cliente, tales como gestores de datos en forma de formularios y listados.

Así mismo, ha sido necesario establecer un servicio de conexión con Amazon y Modalia basado en procesos de sincronización de datos, creación de logs de la comunicación, y un servicio de monitorización del mercado basado en llamadas al servicio y alarmas. Además, como se ha mencionado, todo esto se ha realizado en dos capas. Primero, se realizará un desarrollo genérico con el modelo de datos de la empresa, en lo que llamaremos *el producto*, y, sobre ella, a través de métodos de desarrollo tales como herencia de objetos, interfaces y llamadas a subrutinas, se ha desarrollado la personalización propia para Alma en pena.

Para el desarrollo del modelo de datos se han tenido que crear procesos diarios de sincronización entre la base de datos (DB) del cliente y la propia, y un proceso de conexión paralelo, también bajo demanda, con el sistema de Amazon Marketplace, que se realizará mediante demanda propia del cliente o mediante sincronizaciones periódicas.

Así mismo, ha sido necesario centralizar el proceso de venta, para conseguir una mayor eficiencia de los procesos. Previamente el sistema de gestión de ventas se llevaba desde sus servidores y Yunbit se limitaba a reportar las ventas del comercio electrónico desarrollado previamente por Yunbit. Para solventar esta restricción se ha realizado una ampliación de los servicios ofrecidos para tener disponibles en nuestros sistemas tanto las ventas de Ecommerce, las de las tiendas físicas, reportadas por Modalia, y el stock y ventas correspondientes a Amazon, para poder gestionar los pedidos, las ofertas y el stock de una manera más sencilla y sin provocar inconveniencias a los consumidores finales.

1.3 *Propósito*

Este proyecto tiene como propósito, por tanto, desarrollar un sistema que sea capaz de gestionar los productos, el stock y los pedidos de un comercio electrónico y permitir la venta sobre Amazon de manera automática, sin que el cliente tenga que entrar a gestionar sus ventas o productos en dicha plataforma, sino que lo haga a través del gestor proporcionado por Yunbit, teniendo como objetivo final la presencia de nuestro cliente en Amazon, lo cual aumentará su número de ventas, visibilidad y beneficios, debido a la importancia de Amazon en el comercio a través de internet.

Para ello se han creado tareas automáticas a través de procesos en PHP, llamados con el *crontab* de los servidores Linux de Yunbit, que permiten, primero, la sincronización diaria de los productos de Alma en pena (A través de un servicio llevado por una empresa externa, Modalia) a través de un servicio previamente desarrollado pero ahora ampliado, y

una posterior sincronización a través de otro servicio desarrollado en PHP con los servidores de Alma en pena, llevando una comunicación basada en el protocolo HTTP con mensajes y respuestas en XML. Se adjuntan en el anexo C algunos de los XSD que se han usado para validar dichos mensajes.

La DB se ha desarrollado mediante MySQL, y se controla a nivel de interfaz de usuario mediante gestores de datos desarrollados a través del framework GeaWebBlocks de Yunbit. Mediante dichos gestores, el cliente es capaz de trabajar y exportar los datos sin necesidad de conocimientos de programación algunos, para poder realizar una gestión autónoma de sus propios contenidos. Así mismo, se ha definido un sistema de permisos para distintos usuarios de la plataforma, para permitir trabajar con los datos tanto a los administradores de Alma en pena, que tendrán mayor cantidad de permisos, como a los gestores de contenido, evitando así que un usuario con menor responsabilidad pueda perjudicar el funcionamiento normal de las ventas.

También ha sido creado un sistema automático de alertas para poder avisar automáticamente al cliente por correo electrónico si en Amazon se suben, por ejemplo, los mismos productos por otros vendedores con precio inferior, y así poder mantener la competitividad en el mercado. Estos watchers se ejecutarán una vez al día.

1.4 Organización de la memoria

La memoria consta de los siguientes capítulos:

En la **sección 2** se realiza un estudio del arte donde se puede ver, de manera resumida, la evolución de las tecnologías PHP y MySQL, además de mencionar las usadas en este proyecto, y se llevará a cabo un estudio de la evolución temporal que han tenido los frameworks propios GeaWebBlocks y Businesscloud. Así mismo, se realiza un estudio de distintas herramientas similares que se encuentran en el mercado que permiten la sincronización con Amazon, y se explica por qué se ha considerado el realizar un nuevo desarrollo en lugar de recurrir o contratar las herramientas existentes, y las ventajas que proporciona frente a un servicio de los antes mencionados.

En la **sección 3** se lleva a cabo un estudio de los objetivos del proyecto, y se hace un análisis de las partes de las que constará, así como de sus requisitos funcionales y no funcionales. En esta sección se especifican los resultados esperados por parte de la herramienta en la que se basa este TFG y sus principales funcionalidades.

En la **sección 4** se realiza la definición del proyecto, en la que se explicará la metodología del desarrollo, los procesos y herramientas usadas durante todo el proceso.

En la **sección 5** se puede encontrar un análisis y diseño de la herramienta, la base de datos, los procesos de alarma y sincronización y el sistema de gestión de datos.

En la **sección 6** se centra en los detalles acerca de la implementación del proyecto, donde se puede ver todo el proceso de desarrollo y de creación de logs y registros internos, además de los procesos de la DB y de los demonios que lleven la sincronización.

Durante la **sección 7** se exponen las pruebas y logs que se han llevado a cabo para comprobar el funcionamiento interno de la herramienta. Para ello se va a probar la creación de listings (Listas de productos) de pruebas en Amazon y su compra por el propio usuario, para poder ver su correcto funcionamiento.

Finalmente, en de la **sección 8** se presentan las conclusiones de este TFG, mientras que el trabajo futuro y las posibles ampliaciones se explican en el **apartado 9** de esta memoria.

Por último, en los **Anexos** se muestran partes del código escrito durante este desarrollo, además de los modelos de datos y los XML de sincronización con Amazon, acompañados de sus correspondientes XSDs, usados para la validación de los archivos generados.

2 Estado del arte

2.1 *Introducción*

Para poder ubicar este TFG, es necesario saber en qué consiste un comercio electrónico, y los principales portales de venta electrónica que hay, entre los que se incluye Amazon, del que vamos a ver cómo pasó de ser una tienda de libros online a ser el mayor portal de comercio electrónico del mundo.

Un comercio electrónico, también conocido como Ecommerce, es un nuevo método de venta de productos facilitado por las nuevas tecnologías, que le permite al comerciante poder llegar al cliente a través de internet, ya sean páginas de venta de la marca, como portales que vendan productos de terceros. Entre estos englobamos a Amazon, eBay o Google.

Para este tipo de comercios, se usan formas de pago digitales, como los TPV, que permiten el pago con tarjetas bancarias, o servicios como Paypal. Algunos comercios también admiten transferencias e incluso domiciliación bancaria.

Gracias a este tipo de comercio, las empresas de venta han podido reducir en gran cantidad su coste de distribución y de publicidad, ya que son accesibles a cualquier persona del mundo con acceso a internet, y sólo es necesario asumir costes de almacenamiento, de mantenimiento del servicio web y de logística.

Para empezar, se dará un pequeño viaje a través de las herramientas usadas para el desarrollo del proyecto: PHP y MySQL, y se verán en qué consisten el Framework de la empresa Yunbit que se ha usado para el desarrollo de este producto: webBlocks.

Por último, se verá qué es el Marketplace de Amazon y su importancia, así como las distintas formas de subida de datos al Marketplace de Amazon. Finalmente, se estudiarán algunas herramientas existentes para dicha tarea, y, mediante un estudio de sus particularidades y de los requisitos existentes, se mostrará la necesidad de la implementación de un desarrollo personalizado con el Marketplace Web Service de Amazon.

2.2 Tecnologías usadas

2.2.1 PHP

PHP, que es un acrónimo recursivo de **PHP Pre Hypertext-Processor**, es un lenguaje de programación web para uso desde el backend, y puede ser incrustado en HTML siempre y cuando sea interpretado por un servidor. Es el lenguaje de programación de backend más popular, llegando a una cuota, según el estudio realizado por w3tech¹ en 2015, del 82.2% de todos los websites en internet. Entre los sitios más destacados que usen PHP podemos resaltar Facebook.com, Wikipedia.org, o Twitter.com.

Entre los factores que han contribuido a su gran importancia en el mercado del software actual, podemos destacar el que sea un lenguaje publicado bajo licencia de software libre, su amplia compatibilidad, dado que puede ser despegado en la mayoría de servidores y en casi todos los sistemas operativos y plataformas de forma gratuita, mediante servidores web tales como Apache. Entre sus rasgos funcionales, podemos destacar su bajo consumo de recursos o su elevada potencia, al ser un lenguaje de programación de bajo nivel, comparable con Perl o C.

Sin embargo, no siempre fue tan conocido ni importante este lenguaje. Su origen se remonta a 1995, cuando un groenlandés, *Rasmus Lerdorf*, realizó un CGI (interfaz de entrada común) en Perl que llamó Personal Home Page, para estudiar las estadísticas de acceso a su página web. Recibió tantas solicitudes para obtener dicho programa, que lo recopiló en una versión estándar inicial que llamó **PHP Tools** y liberó el código el 8 de Junio de ese mismo año. Desde entonces, al ser software libre, ha evolucionado de forma ‘descontrolada’, sin embargo, siempre ha habido una línea estándar de desarrollo de este lenguaje, que desde 2001 es controlada por *The PHP Group*, que revisa y lanza nuevas versiones de PHP, yendo por la 7.1 en el momento de la redacción de este trabajo.

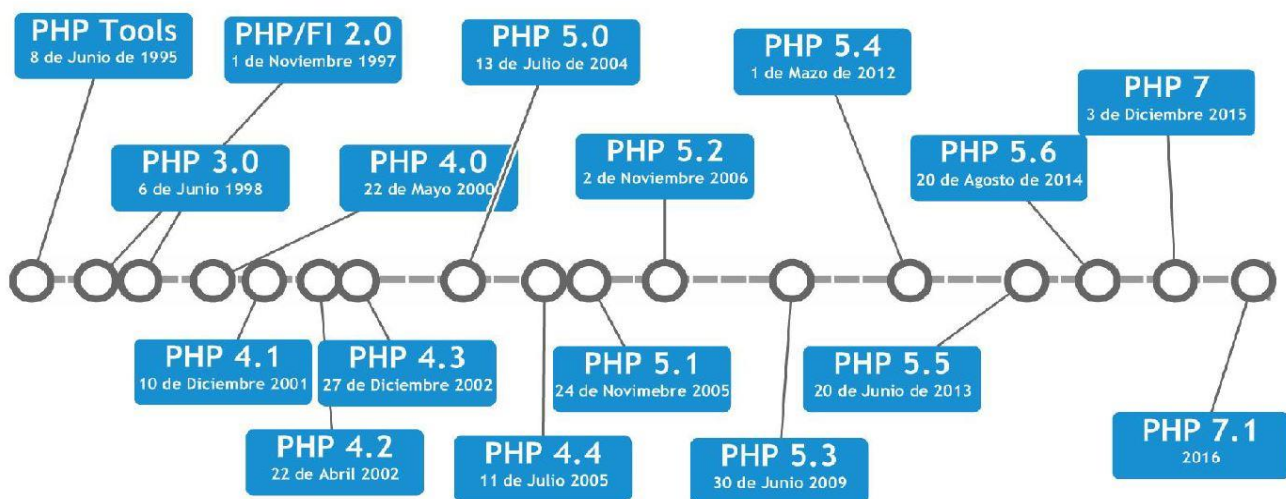


Figura 1 : Línea temporal de PHP

En la figura 1, se puede ver una línea temporal resumiendo los principales eventos en el desarrollo de PHP como lenguaje de programación, incluida la versión utilizada para el desarrollo de esta herramienta, la 5.5.

Cabe destacar una anomalía. Como podemos ver, salta de la versión 5.X a la 7, obviando la 6. Esto se debe a que PHP6 nunca vio la luz, ya que gracias a la descoordinación de los desarrolladores del grupo PHP no se consiguió el desarrollo de una versión estable, por lo que, dado el fracaso, se volvió a partir de la versión anterior para obtener la inmediatamente posterior, que, para evitar problemas de confusión con su ‘hermana’ fallecida, se la denominó PHP².

2.2.2 MySQL

MySQL, acrónimo de *My Structured Language Query*, es un Sistema de gestión de bases de datos relacionales. En Mayo del 2016³, fue el segundo modelo de bases de datos más usado, superado únicamente por Oracle, y cuenta con un crecimiento constante de usuarios, al contrario que este, que tuvo un retroceso de 5 puntos, además de ser el modelo de bases de datos relacionales de código abierto más popular. Entre las páginas más conocidas que usen este modelo de base de datos, podemos encontrarnos con Youtube, Facebook, Wikipedia o Google.

En su gran mayoría, MySQL está escrito en C y C++, y su parser de consultas está escrito en yacc, además, incluye un analizador léxico propio. Dada su importancia, está incluido por defecto en los frameworks web más populares, como son LAMP (WAMP Y MAMP inclusive) o XAMPP, además de en diversos frameworks como Drupal o WordPress.

Este modelo fue diseñado por la compañía *MySQL AB*, creada por *David Axmark, Allan Larsson y Michel Widenius*. La primera versión pública de **MySQL** fue la 3, lanzada el 23 de mayo del 1995, surgida como un sustituto de la obsoleta *mSQL*, pero no fue hasta el 8 de Enero de 1998 que llegó a Windows 95 y NT. A día de hoy, está vigente la versión 5.7 de MySQL, liberada el 21 de Noviembre de 2015, y durante toda su vida, sean ido mejorando las distintas capacidades de este tipo de base de datos, ante todo de rendimiento, pero también se incorporaron nuevas herramientas, como el uso, cada vez más avanzado, de tablas de datos transaccionales.

Para este proyecto se ha usado la edición 5.6 de MySQL, que fue de dominio público a partir del 2 de Febrero del año 2013. En la figura 2, podemos ver resumidamente la vida que ha tenido MySQL desde su creación hasta el día de hoy⁴.

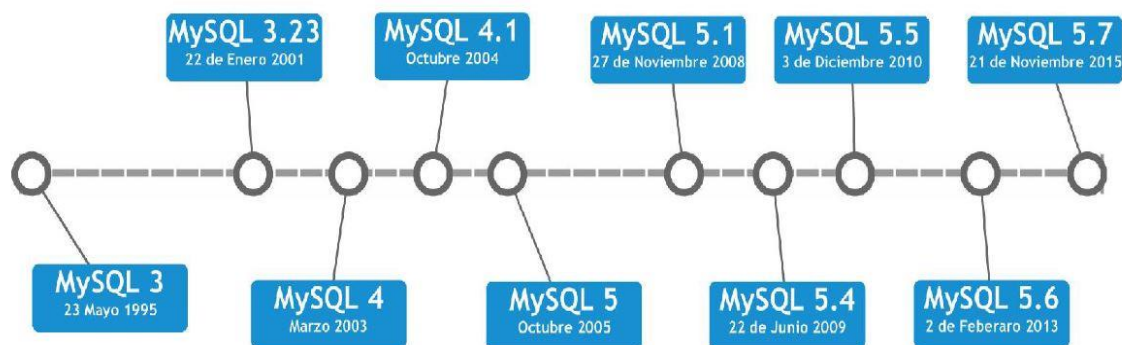


Figura 2: Línea temporal de MySQL

2.2.3 GeaWebBlocks

Para los desarrollos internos, la empresa Yunbit utiliza un framework propio para simplificar y unificar la creación de software, facilitando así el soporte y mantenimiento del código.

Este framework ha sido desarrollado siguiendo un modelo de capas y redirectores. Consiste en el uso de redirectores y archivos de parametrización. Fue inicialmente pensado para que se pudieran realizar cambios de la lógica mediante simples cambios de parámetros, pero acabó convirtiéndose en un framework completo.

La estructura de esta herramienta se divide en 4 capas: *configuración*, *autenticación*, *lógica* y *presentación*. Todo esto se hace mediante la llamada de redirectores, que, dependiendo de cuál sea, permite la creación de gestores de datos o de frontend, o la inclusión o no inclusión de plantillas (la capa de presentación). Así mismo, es posible omitir la capa de autenticación, aunque es recomendable tener otro sistema de acceso a los datos si se omite, para evitar problemas de seguridad en el sistema.

Esta tecnología es la que se ha usado para desarrollar los gestores de datos para que el cliente pueda controlar todos los datos de su producto de forma sencilla a través del backend que tiene contratado con la empresa Yunbit.

Los archivos en la raíz contienen las llamadas a los redirectores, y, como en la siguiente imagen, se pueden encontrando varios directorios con diversa funcionalidad:

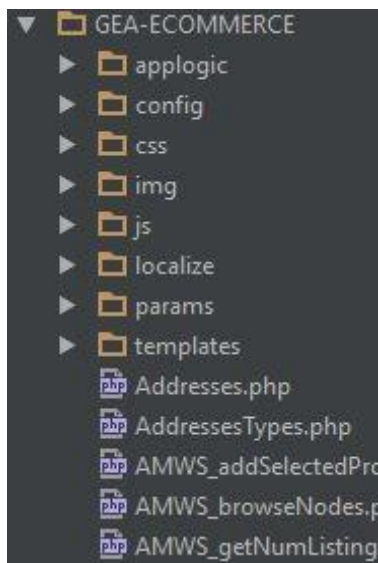


Figura 3: Directorios de WebBlocks

- **Applogic.** Contiene la lógica.
- **Params:** Este directorio contienen los archivos de parametrización para la creación de gestores.
 - **Localize:** Esto permite la inclusión de diccionarios, para el multiidioma.
 - **Css, img y js:** Contendrán los archivos .css, de imágenes o javascript usados.
 - **Templates.** En este archivo se guardan los templates. Por defecto, si no se indica lo contrario, buscará un template con el mismo nombre que el archivo situado en la raíz.
- **Config:** Contendrá los archivos de configuración, destacando el local_globals.php, que contiene toda la configuración del cliente.

El sistema de templates que utiliza es muy similar a uno de los más extendidos en php, el PHPbb, basándose en la versión 2 de este framework para su capa de presentación⁵.

2.3 Amazon Marketplace Web Service

Como ya se ha comentado, Amazon es el mayor portal de venta online a nivel global. En un principio, sólo permitía la subida de sus productos a partir de su gestor, o mediante la creación de archivos de carga y subiéndolos a su web, pero en 2009 lanzaron una API pública⁶ para permitir la descentralización de estas subidas.

Esta API recibió el nombre de Amazon Marketplace Web Service⁷, y es un servicio web integrado con Amazon que permite a los comerciantes intercambiar datos de los productos, envío, pedidos, stock, informes, etc. mediante métodos de programación y llamadas HTTP. Estos servicios permiten automatizar el servicio de ventas, aumentando la eficiencia de las ventas y ayudando a los vendedores a mejorar su negocio.

Este servicio se basa en una comunicación cliente-servidor con los servidores de Amazon, donde, mediante llamadas HTTP y mensajes en XML, se puede realizar todo tipo de carga que fuera posible de manera manual y además, nuevas opciones como obtener informes de precios, productos, venta o competencia.

Estas funciones dan acceso a los centros logísticos de Amazon, donde el vendedor puede decidir llevar su catálogo (pagando una tasa por almacenamiento) y que Amazon se encargue de la distribución o, o, si ya tenía un sistema de logística previo, como es el caso de Alma en pena, puede encargarse él mismo de la distribución de los productos.

Para este fin, Amazon proporciona una extensa documentación, que permiten la llamada a estos servicios. Dichas llamadas, como se explicará más tarde durante el desarrollo, están encapsuladas en sus *Envelopes*, y tienen restricciones de tamaño y de servicio, usando un sistema de Throtting para rechazar los paquetes cuando recibe muchos de un mismo vendedor.

La API de Amazon MWS está dividida en las siguientes secciones:

- Feeds
- Informes
- Pagos
- Envíos entrantes
- Inventario
- Envío salientes
- Pedidos
- Productos
- Recomendaciones
- Vendedores
- Suscripciones

Para subir datos y productos a Amazon, existen varias opciones, que vamos a ver a continuación, tanto propias de Amazon como herramientas de terceros, pero y vamos a ver por qué fue necesaria la realización de este desarrollo en vez de usar herramientas externas.

Primero vamos a ver el gestor de archivos de Amazon. Este permite, mediante la subida de diversos archivos de tipo Excel, el poder subir productos a Amazon sin tener que estar creándolos como si se tratara de un gestor de contenidos.

Para poder llevar a cabo dicho procedimiento⁸ es necesario, primero, descargarse el fichero template desde el gestor. Resaltamos, además, que es necesario un tipo de plantilla por cada tipo de producto, por ejemplo, unas *Alpargatas* usarían un tipo de plantilla distinto que unas *Botas*.

Una vez obtenido el archivo, podemos acceder a diversas pestañas de un archivo Excel, que incluyen un manual para rellenar la plantilla, normativas y legislaciones. Podemos ver que el Excel tiene 147 columnas que hay que completar con los datos del producto para poder subirlo.

Tras rellenar el archivo, se sube a la página de Amazon, donde nos genera un informe de estado, que contiene el estado de la subida y si ha habido errores, y que se debe de consultar para poder corregir los mismos.

Entre las herramientas de terceros que automatizan la subida de Amazon, podemos encontrarnos con Magento, Shopify y ShoppingCartElite.

Estas tres tiendas permiten subir, mediante la API de Amazon, automáticamente los productos que tienen en stock.

2.4 Conclusión

La funcionalidad que se quiere desarrollar es bastante similar a las de estas tiendas, pero con una mejora de gran relevancia. Esta mejora consiste, esencialmente, en que desde el comercio electrónico gestionado por Yunbit, se permita sincronizar las cargas de productos desde Modalia con las subidas a Amazon, y las ventas desde las

tres posibilidades (tiendas físicas, gestionadas por Modalia, tienda electrónica, gestionada por Yunbit, y Amazon) para gestionar el stock y distribución de los productos desde diversos almacenes.

Esto supondría una gran disminución del trabajo por parte de la empresa cliente, ya que no necesitaría gestionar y mantener sincronizados manualmente cada sistema de ventas, ni gestionar los pedidos desde Amazon, cosa normal con el resto de aplicaciones, ya que sería la propia herramienta desarrollada la que automatizaría todo este proceso.

Así mismo, tendría una repercusión económica y tecnológica en Yunbit, al convertir dar un gran empujón a su Ecommerce dentro del mercado, dando un método de distinción ante la competencia y aumentando en gran parte la potencia de su producto, facilitando en el futuro nuevas ventas a otros clientes.

Por ello, se ha visto como una significativa mejora del producto de Ecommerce de la empresa Yunbit el desarrollo de esta funcionalidad de sincronización con Amazon, por lo que, tras una reunión para presentar este posible desarrollo, ha sido aprobada por el CEO y CTO de la empresa, Roberto López y Mario Villar. Esta mejora y todo lo que ella supone se describirá en detalle a lo largo de esta memoria.

3 Definición del proyecto y objetivos

Como ya se ha mencionado, la finalidad de este proyecto es desarrollar un módulo para el producto de Yunbit que permita la sincronización e integración del sistema de inventario y ventas del portal de Amazon, a través de la API de su Marketplace Web Service, y todo este proceso se automatice siendo transparente para el usuario. Así mismo, se debe de integrar este módulo con el Ecommerce del cliente Alma en pena, desarrollado por Yunbit (se puede acceder a la tienda a través de <http://www.almaenpena.es>) y adaptar su lógica al sistema de gestión de contenidos personalizado que tenía Alma en pena.

Así mismo, cabe destacar que este módulo se ha hecho de forma genérica, utilizando métodos como herencia de objetos y parametrización, para que otros futuros proyectos de Yunbit puedan implementar este módulo, con todas sus funcionalidades, de forma sencilla.

Inicialmente, el sistema permitía gestionar el estado de los pedidos y sincronizar el catálogo de productos con su anterior gestor de contenidos, que aún usan para la distribución y el stock de las tiendas físicas que venden zapatos de Alma en pena. Este es el sistema que se usa como punto de partida de este trabajo.

A continuación, se van a listar en forma de resumen los objetivos y las funcionalidades del proyecto objeto de este trabajo, y, durante la sección, se explicarán los diversos puntos de forma más detallada.

- *Sistema de objetos para la estructura* de productos de Amazon y la generación de sus mensajes.
 - Creación de un modelo único de datos para los productos, precios y stock, tanto como para la Tienda de Yunbit como para la de Amazon.
 - Desarrollo del sistema de comunicación con su servidor: Envío de peticiones y recepción de respuestas.
- Ampliación del *sistema de datos preexistente*.
 - Ampliación en los datos del producto
 - Desarrollo de los pedidos y ventas: Gestionar los pedidos de Alma en pena desde el gestor de Yunbit, en vez de Modalia.
- Desarrollo de gestores propios para que el cliente pueda gestionar los listings, productos y precios del sistema.
- Tareas de sincronización automáticas que automaticen el proceso de actualización y refresco de los datos de Amazon.
- Watchers de Amazon para avisar de eventos.
 - Avisos de cuando la competencia mejora el precio de tus productos.

Como se parte de un proyecto ya existente, es necesario mencionar lo que está hecho y lo que el estudiante ha desarrollado, o modificado de forma específica. El sistema de gestión de la tienda de Alma en pena desarrollado previamente por Yunbit previamente a este trabajo permitía:

- Hacer una llamada (por demanda) a Modalia para obtener su catálogo de productos.
- Cuando se hacía un pedido en la tienda, realizar una llamada a Modalia para enviarles los datos del pedido y de la distribución.

- Gestores básicos de productos (Stock, colores, tallas y descripción).

De estas funcionalidades, el alumno ha modificado lo siguiente:

- Ha convertido la llamada al catálogo en un proceso de sincronización diario (para controlar el stock y los cambios en el catálogo).
- Ha añadido nuevos campos de control en los pedidos para guardar los datos de Amazon y distinguir estos pedidos del resto (al tener distinto flujo de comportamiento).
- Ha ampliado la estructura de datos de los productos para recoger los datos de Amazon y añadir nuevos campos de características.
- Ha modificado las tareas de envío de pedidos a Modalia.

El resto de las funcionalidades y servicios descritos durante toda la memoria han sido implementados desde cero por el alumno, en particular:

- Todo el modelo de datos referente a los productos y listings de Amazon.
- Todo el sistema de gestión de listings.
- Las tareas de sincronización de productos con Amazon.
- Las tareas de creación y flujo de vida de pedidos de Amazon.
- Nueva tarea de flujo de vida de pedidos para Alma en pena.
- La tarea de seguimiento de precio de los productos.

A continuación se va a ver, en líneas generales, cada parte de lo que supone esta aplicación. Se comienza explicando el modelo de datos de Amazon y cómo se ha integrado con el de Yunbit, a continuación se verá el modelo de datos usado y se explicará en qué consiste la aplicación desarrollada.

3.1 Modelo de datos

Para la comunicación con Amazon, en su API tienen un modelo de datos que ha sido necesario imitar, en forma de interfaz, para poder realizar la conversión entre ambos modelos de datos: el de Alma en pena y el de Amazon. El modelo de Amazon viene definido por unos XSDs (XML Schema Definition), concretamente 74 XSDs, aunque dado que muchos son por tipos de productos, como Juguetes o Joyería, no ha hecho falta analizarlos todos. En el Anexo C podemos encontrar un listado con los XSDs usados. De forma resumida, se va a explicar en qué consiste su modelo de datos y qué se hará como interfaz de modelos.

3.1.1 Productos

Amazon utiliza un sistema de herencia para la gestión de sus productos. Mantienen dos clases de productos: Padres y Variaciones. Los padres serían, por así decirlo, el nodo raíz, y las variaciones, que se definen por tipo de variación (talla, color, talla y color, material, etc.) son los distintos nodos que van colgando, mostrando al final a la venta sólo los nodos hoja de esta relación. Todo esto se mostrará con más detalle durante el análisis de datos.

Así mismo, cada producto tiene varios tipos de características, en las que se destacan, por ahora, los detalles, que contienen información sobre el producto en sí (país,

fabricante,...) y tipo, en el caso de este proyecto, zapatos, que a su vez contiene otras características propias.

Amazon tiene como entidades separadas los Productos, Imágenes, Precios, Inventario (stock) y Relaciones (quién es variación de quién), por lo que también se desarrollarán objetos para gestionar estas estructuras de datos y su respectiva creación y validación de los XMLs.

Para lidiar con esto, se ha hecho un sistema de objetos genéricos que, tras hidratarlos con datos, crean el archivo XML necesario para comunicarse con Amazon. Así mismo, se ha creado una clase ‘Cliente’ para gestionar la comunicación de paquetes y el reporte de errores en el procesado de los datos por parte de Amazon.

3.1.2 Pedidos

Para la gestión de pedidos de Amazon, lo que hace Amazon es guardar los pedidos y sólo reportarlos si se solicitan vía API, por lo que ha sido necesario crear una tarea programada cada hora que pide los pedidos, con los estados deseados, desde la última ejecución a Amazon.

Una vez que se tengan los datos de pedidos, se contrastará con el estado de los productos, (stock, disponibilidad logística) y se confirmará o rechazará el pedido. Hecho esto, Amazon confirma la venta al cliente y, dado que Alma en pena ha decidido mantener su sistema propio de distribución (existiendo la opción de darle los productos a Amazon y que lo gestionen ellos), se realiza en envío a través de su propio sistema de logística.

3.1.3 Cliente

Como ya se ha mencionado, se ha creado una clase Cliente que gestione la creación de paquetes de datos y la comunicación con Amazon, y este cliente, dado el carácter genérico de este módulo, se personalizará (mediante un sistema de herencia e interfaces) en cada aplicación, tal y como se ha hecho en esta implementación.

Dicho cliente contiene las tareas de:

- *Creación de paquetes de datos.*
- *Gestión de tareas de envío de Listings.*
- *Generación de informes de procesado de datos tras su envío.*
- *Gestión del tráfico de envío:* Amazon MWS tiene un sistema de throttling (escalado de frecuencia de recepción de paquetes) que limitaría la cantidad de mensajes que puede recibir en un determinado tiempo.

El cliente se encargaría de, si se recibe mensaje de error por throttling, reintentar el envío pasado un tiempo y, de volver a ocurrir, repetir este proceso aumentando de manera exponencial el tiempo de espera.

3.2 Gestores y modelo de datos

Durante el desarrollo, se han visto limitaciones en el antiguo modelo de datos, dado que anteriormente sólo se gestionaban las descripciones, tallas, colores, precios y stock. Por lo tanto, se va a realizar una ampliación del modelo de datos, permitiendo poder gestionar los diversos atributos permitidos por Amazon

3.3 Tareas de sincronización

En esta parte se definen las distintas tareas de sincronización, tanto automáticas como manuales, que sirven para conectar con los distintos servidores y automatizar todos los procesos, dando valor añadido a la aplicación al aliviarles la tarea a los gestores de contenido.

3.3.1 Sincronización de productos

Para poder gestionar todos los servicios de venta, ha hecho falta sincronizar el stock y los datos de productos y promociones entre los 3 servicios que pasarían a existir: el de venta físico, gestionado por Modalia, el comercio electrónico de Yunbit y el Marketplace de Amazon.

Para ello, se han diseñado unas tareas de sincronización consecutivas, con el objeto de centralizar la información de stock, productos y precios, lo que da la puerta, en un futuro, a Alma en pena para finalmente no depender de Modalia y contratar a Yunbit también para la gestión física.

Estas tareas se han dividido en dos etapas: primero, se le piden a Modalia los datos del catálogo, los cuales se almacenarán en la base de datos de Yunbit, y tras eso, se detecta si ha añadido nuevos productos o han habido cambios en los existentes. De haber habido cambios en los productos o stock que están listados en Amazon, se marcan los productos actualizados para que en la sincronización con Amazon se actualicen.

La segunda fase consiste en una tarea automática ejecutada tras esta primera, que coge aquellos productos en los que se han producido variaciones de stock o alguna modificación, ya sea a través del gestor o de la anterior tarea, y los envía a Amazon para actualizar los datos de dichos productos. Esta tarea, así mismo, da de alta a los nuevos listings que se hayan publicado, de haberlos.

3.3.2 Carga manual

Para poder solucionar errores temporales o no esperar a la ejecución de una tarea de sincronización, se ha creado una funcionalidad de activación manual para la sincronización de listings. Esta tarea permite reemplazar los datos existentes en Amazon sobre un listing con los datos presentes en la base de datos en el momento de su activación.

3.4 Reportes y alertas

El sistema de Amazon tiene una gran ventaja respecto a otros servicios de venta, incluyendo la propia tienda de Alma en pena, y es que tiene un gran volumen de visitas y, debido a eso y a la gran cantidad de vendedores, puede obtener estadísticas bastante útiles para el mercado.

Como funcionalidad básica en este proyecto, se ha desarrollado un único tipo de reporte, aunque, en futuro trabajo, se estudiará la ampliación de la sección de reportes, y el servicio que se implementará será un estudio de los precios de mercado. Cada día, pedirá a Amazon el precio de los productos activos en su Marketplace, pero de distinto vendedor, y generará un reporte por correo en caso de que algún producto sea superado (ya sea por oferta o por precio básico) en precio, para que Alma en pena pueda fijar ofertas o cambiar los precios para conseguir ventaja frente al resto de competidores.

4 Análisis y Diseño

4.1 Análisis

Durante esta sección del documento, se va a ver un análisis previo al diseño del sistema, donde se van a estudiar los componentes de todo el sistema y por último se obtendrá un listado de los requisitos funcionales y no funcionales.

4.1.1 Pedidos

Los pedidos de Amazon se realizan en el propio portal de ventas, pero dan un margen a la validación por parte del mercader para cancelar el pedido, por falta de stock, por ejemplo, o al usuario para arrepentirse.

El ciclo de vida de un pedido de Amazon es el siguiente:

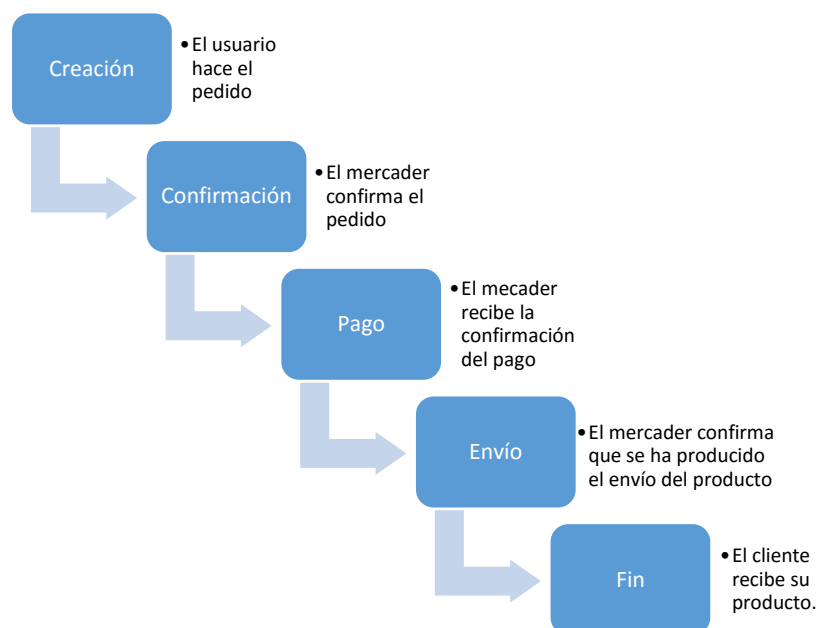


Figura 4 : Ciclo de vida de un pedido

En este ciclo, se pueden producir distintos puntos de ruptura:

1. Cancelación por parte del cliente [Entre Creación y Confirmación].
2. Rechazo del pedido por parte del mercader [En confirmación o Envío].
3. Error al realizar el pago del producto.

En cada uno de estos casos, Amazon marcará como rechazado el pedido. Si no, seguirá el curso normal y tendrá que ser el mercader, Alma en pena en este caso, el encargado de hacer llegar el pedido al comprador.

Sin embargo, esto tiene una dificultad, y es que Amazon no informa en ningún caso del estado del pedido. Para ello, ha hecho falta desarrollar una tarea programada (mediante el crontab) que cada X tiempo (se ha programado cada 30m) realice una petición a Amazon para obtener todos los estados de los pedidos.

Si ocurre un pedido o uno activo es cancelado por parte de Amazon, mediante esta llamada se actualizaría el stock de Yunbit, ya que Amazon actualiza su stock a tiempo real, y se enviaría el pedido a Alma en pena para que pueda o cancelar la tramitación, o iniciar el proceso de envío. A continuación podemos ver el ciclo de vida de un pedido de Amazon (Imagen obtenida de la API de Amazon):

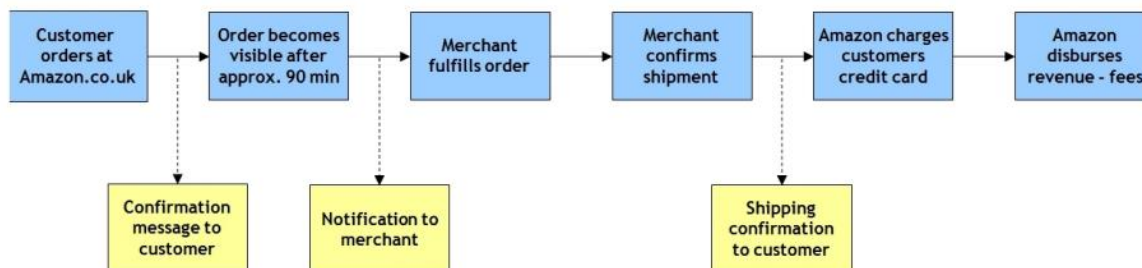


Figura 5: Vida de un pedido, desde que se crea hasta que se realiza el cobro

Así mismo, cada vez que hay una venta por parte del servicio web de Yunbit, es necesario sincronizar el stock de Amazon para evitar un desequilibrio. Esto se verá más adelante.

4.1.2 Productos

Amazon usa un sistema de productos bastante singular. Ya se ha mencionado con anterioridad en la memoria, así que en este apartado sólo se va a ver en lo concerniente al análisis previo al diseño.

El sistema que utiliza Amazon para representar sus productos, se puede ver como un árbol de herencia Padre-Hijo, o, en su nomenclatura, **Parent – Variation**. Los productos padre son sólo un nodo abstracto del que penden las variaciones, las cuales representan el producto final al que tiene acceso el cliente de Amazon.

Dichas variaciones tienen un tipo y un atributo, por ejemplo, las variaciones de tipo 'Material' tienen el atributo de 'Material', y son únicas: no pueden haber dos variaciones con el mismo atributo y tipo que pendan del mismo padre. Para este proyecto interesan las variaciones de tipo 'Color-Talla', ya que son las que se usan en el negocio de la venta de zapatos, pero se contempla en un futuro, para otros proyectos, usar otro tipo de variaciones.

Así mismo, de cada producto hijo (también se pueden usar a los padres para esta relación, aunque, ya que son invisibles, es inútil) penden varios tipos de atributos, aparte de las descripciones incluidas en el producto. Estos son: **Relaciones, Imágenes, Stock y Precios**.

- **Relaciones:** Establecen la relación Padre – Hijo. También pueden servir para dar relaciones de similitud entre productos, aunque no entra en el alcance de este proyecto.
- **Imágenes:** Contiene las imágenes de cada producto.
- **Stock:** Indicará el número de unidades, si hay varios almacenes indicará a cual se refiere, y permite otras opciones de distribución, como que Amazon distribuya los productos.
- **Precios:** Contiene la información del precio standard del producto y de las distintas promociones, con fecha de inicio, de fin, y número máximo de unidades disponibles para la promoción.

Todo esto se verá con más detalle en la sección de diseño, al ver el esquema UML (Figura 10) del sistema de objetos que ha desarrollado el alumno para la gestión de los productos.

4.1.3 Roles de usuarios

Para este sistema de comercio electrónico, ha sido necesario definir varios tipos de usuarios y permisos, ya que no se va a encargar una única persona de gestionar todo esto, al ser Alma en Pena una tienda grande (tiene millones de euros en facturación anual).

Los siguientes tipos de usuarios van a ser necesarios:

- **Gestor de contenidos:** Este usuario realizará la gestión de los datos de los productos, y podrá modificar desde el gestor los datos propios del producto (descripciones, atributos, imágenes, precios). Este usuario podrá dar de alta nuevos listings y modificar los existentes, pero no podrá borrarlos.
- **Gestor de ventas.** Este tipo de usuario tendrá acceso a la gestión de pedidos, y podrá revisar y modificar los datos de los pedidos, y crear facturas a partir de los pedidos. Este perfil ya existía en el sistema de Yunbit.
- **Administrador.** Serán los usuarios con privilegios globales, que puedan acceder a todas las funcionalidades, podrán borrar productos, y serán los únicos que puedan borrar listings y tener acceso a los logs técnicos de errores. Es el tipo de usuario de la cuenta de soporte de Yunbit y del Manager de Alma en pena.

4.1.4 Requisitos funcionales

A continuación se listan los requisitos funcionales, y en la siguiente sección los no funcionales, más relevantes:

Código	Descripción
RF01	El sistema permitirá al usuario crear listados de diversos de productos,
RF02	En cada listing, el usuario podrá tener una variación distinta sobre el precio base de los productos
RF03	Los listings podrán tener distintos estados (Inactivo, Pendiente de sincronizar, Sincronizado, Con errores, Con alertas) y fechas de activación y expiración, sólo manteniéndose en línea con Amazon

	aquellos publicados y no marcados como inactivos.
RF04	El usuario podrá desactivar productos dentro del listing, desactivándose estos de Amazon.
RF05	Un usuario podrá desactivar un listing, desactivándose todos los productos que no estén en otros listings activos.
RF06	Un usuario podrá filtrar entre todos los productos existentes y seleccionar aquellos que quiere agregar a cada listing.
RF07	Cuando se borre un producto <i>sincronizado</i> del listing, se borrará de la tienda de Amazon.
RF08	Cuando llegue la fecha de publicación de un listing, éste se publicará sin intervención del usuario.
RF09	Diariamente se deben sincronizar todos los productos para evitar un desbalance de stock y actualizar los cambios hechos por el usuario.
RF10	El usuario podrá editar los datos del producto desde la plataforma, creando nuevas características (Material, Talla) de ser necesario.
RF11	Las ventas de Amazon se darán de alta en la plataforma para que el cliente las tramite.
RF12	Si se produce una cancelación de una venta de Amazon, se debe de cancelar el pedido para evitar el envío.
RF13	Si el usuario cancela un pedido desde la plataforma (por falta de stock, por ejemplo), el sistema indicará a Amazon que se ha producido la cancelación.
RF14	El sistema generará mediante una tarea diaria un informe sobre la competencia que enviará al usuario, con un análisis del precio de cada producto en los distintos vendedores de Amazon.
RF15	Tras cada sincronización diaria, se enviará al usuario un informe con el número de productos sincronizados correctos e incorrectos, e información de cómo solventarlos si es problema de los datos.
RF16	Este mismo informe se enviará a los administradores de la aplicación (al alumno y a la cuenta de correo de soporte de Yunbit) con datos del log de errores para su correcta resolución.
RF17	Cuando Modalia realice el envío del producto al cliente, el sistema informa a Amazon mediante un mensaje de cambio de estado del pedido a Enviado (Shipped)
RF18	Cuando se realiza el cambio de estado a enviado, el sistema informa mediante un email al cliente de que el producto está en camino, dando código de seguimiento del envío y tiempo estimado de llegada.
RF19	A través del gestor de listings y productos de listings, el usuario podrá acceder a los productos y, de tener permisos, modificar la información de los mismos directamente.

4.1.5 Requisitos no funcionales

Código	Descripción
RN01	Debe haber tres tipos de permisos para los usuarios: Gestores de marketing, Gestores de contenido, y Administrador. Los gestores de marketing podrán editar listings, los gestores de contenido podrán modificar los datos de los productos mientras que los administradores

	tendrán permisos globales.
RN02	Los productos se agregan de forma responsive mediante javascript
RN03	Se identifica el estado de los productos y listings mediante colores, siendo amarillo el NO SINCRONIZADO, rojo el de ERROR, naranja el de WARNING y verde el de OK.
RN04	Se guardará un log de los mensajes intercambiados con Amazon al que sólo tendrá acceso el administrador
RN05	Mediante un sistema de filtros (Categorías, Temporadas, Activos e inactivos), se pueden filtrar los productos del listado de productos.
RN06	El sistema de subida de productos es capaz de soportar un servicio de filtrado de peticiones al servidor de tipo Throtting,
RN07	Automáticamente el sistema realiza una sincronización de <u>toda</u> la base de datos de Amazon con la de Yunbit, y la de Yunbit con la de Modalia cada día.
RN08	Se mantiene una comunicación de tipo Polling, con mensajes cada hora, con Amazon para verificar el estado de los pedidos.
RN09	Se debe de proporcionar a los usuarios información básica para que puedan operar con normalidad, sin conocimiento técnico de los errores. Para ello se enviará el informe de sincronización diaria.
RN10	Se debe de proporcionar a los usuarios de tipo administrador información más avanzada sobre los errores.
RN11	En caso de que haya un fallo de sincronización, hay una opción de sincronización manual de listings sólo visible para los administradores.

4.2 Diseño

Tras analizar los requisitos, tanto funcionales como no funcionales de la aplicación, y haber estudiado anteriormente el alcance del proyecto y sus objetivos, se va a proceder a explicar el diseño de esta aplicación.

En el siguiente gráfico, se puede ver el enramado de servidores y bases de datos del producto. Lo que se ha establecido, es que la base de datos de Yunbit, la central, pida a la de Modalia los pedidos de las tiendas físicas y cada día le pida su catálogo para actualizar el stock. El usuario accederá a este servidor, el de Yunbit, y podrá crear nuevos productos o modificar los ya existentes, y se mostrarán en la tienda web de Yunbit, donde los clientes pueden realizar pedidos. Ahora, además, se ha instaurado un tercer servidor, el de Amazon, con el que se comunicará el servidor de Yunbit para subir los productos y obtener información de los pedidos.

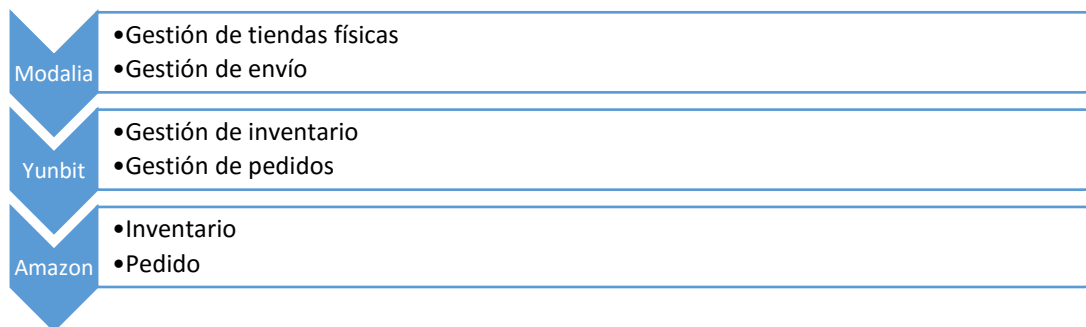


Figura 6: Tareas ejercidas por cada servidor

A continuación se introduce un diagrama (figura 7) que representa la distinta interacción entre los servidores:

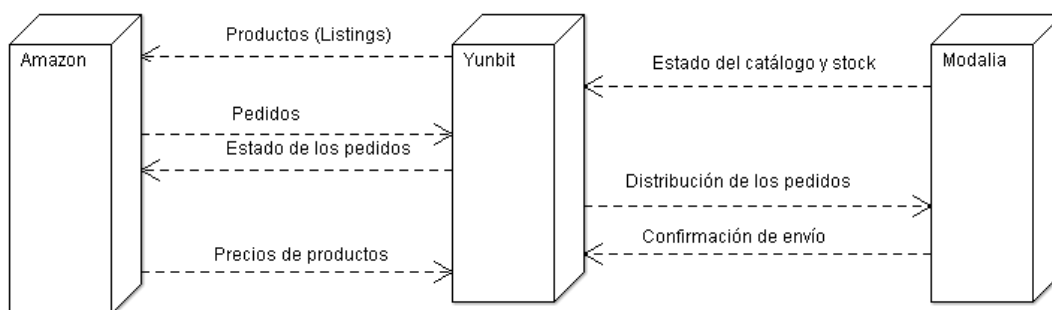


Figura 7: Interacción entre los distintos servidores

A partir de la figura anterior, se puede dividir la estructura del proyecto en las siguientes partes, que se explicarán en los apartados posteriores:

- Base de datos de Yunbit. Esta base de datos (MySQL) contiene toda la información acerca de los productos a la venta, las características, los pedidos, los usuarios, los listings de productos y los logs.
- Gestores de datos. El usuario tiene acceso a un gestor donde, de manera intuitiva, puede ver el estado de la sincronización de los productos con Amazon, y añadir, editar y borrar los listings y productos.
- Sincronización con el servidor de Modalia. Se realiza diariamente una sincronización con el servidor de Modalia para obtener los datos de las ventas hechas en las tiendas físicas, y las actualizaciones en los datos hechos en los productos, a los que se les dará prioridad. Todos estos cambios se deben de sincronizar con los datos presentes en Amazon.
- Modelado de productos. Se han creado objetos en PHP que sirven de interfaz entre el modelo de productos de Amazon y de Yunbit. Estos objetos se hidratan con los datos de la base de datos de Yunbit y permite obtener los mensajes a enviar a Amazon sobre los distintos productos.
- Conexión con Amazon. Se ha creado una clase que sirve de cliente con Amazon. Esta clase se encarga de ofrecer las funciones de sincronización con el servidor

de Amazon, enviar los distintos mensajes, recibir las respuestas, crear los logs con los intercambios de mensajes y actualizar el estado de las sincronizaciones.

- Tareas programadas de sincronización de pedidos y productos. Se ha creado una serie de tareas programadas (se han programado mediante el crontab de los servidores Linux) que llevan la sincronización de los productos y listings [*cada 12h*], la sincronización de los pedidos (de ambos lados) [*cada 30m*], la sincronización de stock [*cada hora*] y la obtención del informe sobre los precios de la competencia [*cada día*].

4.2.1 Base de datos.

La base de datos usada para la integración del Amazon con el Ecommerce de Yunbit es una base de datos relacional MySQL, ampliación de la ya existente para el proyecto Ecommerce de Almaenpena.

Para este proyecto, se han creado nuevas tablas y modificado tablas ya existentes, para poder cumplir con las funcionalidades del diseño. Las nuevas tablas creadas para este proyecto son las siguientes:

- GEAECOM_AMWS_LISTINGS. Esta tabla guarda todos los registros acerca de los listados para subir a Amazon y contiene los datos tales como el nombre, estado del listado, fechas de publicación y expiración, modificación de precio y datos sobre la sincronización.
- GEAECOM_AMWS_LISTINGS_rel_PRODUCTS. Es una tabla relacional que contiene las relaciones 1-n entre el listing y sus productos. Además, tiene el estado de dicho producto (activo/inactivo) y los metadatos acerca de la sincronización.
- GEAECOM_AMWS_NODES. Es una tabla que contiene las distintas categorías de Amazon, que se ha usado en el mapeo entre las categorías de productos de Alma en pena y Amazon.
- GEAECOM_AMWS_LOGS. Esta tabla guarda los datos acerca de la sincronización de productos con Amazon, tales como los tiempos, el resultado y los archivos enviados y recibidos (en formato XML).
- GEAECOM_AMWS_DELETE_PRODUCTS. Esta tabla contiene la cola de borrado. Incluirá los ids de los productos que se borran del gestor de listings y están sincronizados. Con la sincronización se vaciará y se borrará de Amazon. Se hidrata automáticamente mediante Triggers en la tabla de relación listing-producto.

Así mismo, se han modificado las siguientes tablas:

- AP_PRODUCTS. Es la tabla de productos, se le ha añadido datos para el catálogo de Amazon.
- AP_ORDERS. Es la tabla de pedidos, se ha modificado para distinguir entre pedidos normales y de Amazon y asignarles el ID de seguimiento.
- AP_TAGS. Son las categorías de productos, se han modificado para poder mapearlas con las categorías de Amazon.

A continuación se puede ver un esquema con la composición de la DB de los productos. En la siguiente sección vamos a ver un diagrama (simplificado) explicando la estructura de datos desarrollada por el alumno para la DB de Yunbit. Para ayudar a comprenderlo mejor, se ha coloreado la sección de listings con naranja, la de productos con azul, y la de gestión de pedidos con verde:

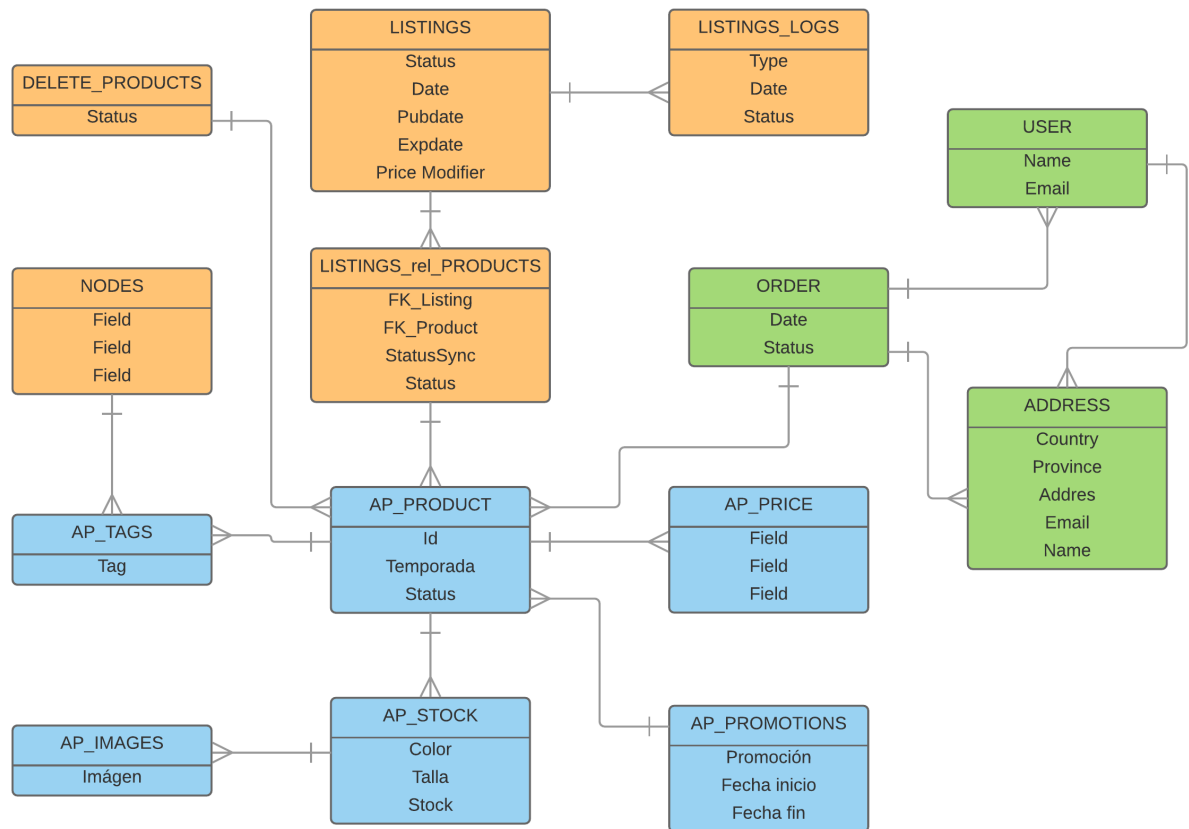


Figura 8: Estructura de la DB

4.2.2 Gestores de datos.

Los gestores sirven como interfaz de usuario para esta aplicación, por lo que se ha buscado que sean intuitivos y fáciles de gestionar, y, mediante un código de colores, a simple vista se pueda obtener información sobre el estado de cada listing y producto. Se han hecho mediante el framework GeaWebblocks de Yunbit, ya mencionado anteriormente en este trabajo, para que mantengan una coherencia con el resto de la aplicación.

4.2.3 Modelado de productos.

Las clases de los productos se han implementado de forma genérica, según el modelo de datos de Amazon, para que pueda ser reutilizado en futuros proyectos basados en este, y cada clase implementa mediante getters y setters las distintas

funciones para tratar los atributos de cada producto.

Esto permite, tras poblar cada objeto con los datos propios de los productos de Alma en pena, poder generar a través de las funciones primitivas de la clase los XMLs propios del envío, además de facilitar la implantación en futuros proyectos.

Mediante un mecanismo similar, la distinción entre genérico y personalización, se ha dividido el diseño de la clase cliente (el que realiza las comunicaciones con el servidor) en dos objetos. El primero, es una clase abstracta que pertenece al producto genérico de Yunbit, y el segundo es una implementación de este, con las funciones necesarias para la personalización de Alma En Pena.

En el siguiente capítulo de la memoria se detallará el modelado de las clases de los productos y clientes.

4.2.4 Sincronización de productos.

La sincronización de los productos se ha planteado de forma que el usuario no tenga que hacer nada especial con respecto a lo que gestionaba antes, más allá de dar de alta los listings de productos.

Para esto, se han realizado una serie de tareas (a través del *crontab*) que periódicamente permiten la sincronización, primero de Modalia con los productos de Yunbit, y luego de estos últimos con los de Amazon.

4.2.5 Sincronización de pedidos.

Dado que el objetivo de este proyecto es realizar una implementación de un Ecommerce con el portal de Alma en pena con la máxima autonomía posible y mínima interacción del usuario con Amazon, se ha tenido que ejecutar en segundo plano toda la lógica necesaria para esta implementación.

Esto se ha conseguido mediante diversas tareas, programadas mediante el *crontab* de los servidores de Linux, que mantendrán una sincronización constante de la base de datos con los servidores de Amazon, consiguiendo, en última instancia, que sea transparente para el usuario.

Además, para lograr esto, mediante el sistema de permisos se ha ocultado la información sobre la sincronización a los usuarios de Alma en pena, manteniendo únicamente el código de color.

Así mismo, mediante este sistema de tareas, se crearán en la base de datos los pedidos hechos mediante Amazon y se gestionaran de manera similar a como se estaba haciendo en la plataforma de Yunbit, permitiendo al usuario de Alma en pena poder cancelar pedidos, así como al de Amazon cancelarlo y que se cancele aquí también. A continuación, se puede ver un diagrama con el flujo de pedidos de desarrollado por el alumno para la sincronización con Amazon y los distintos puntos de ruptura (mediante cancelaciones) de dicho flujo:

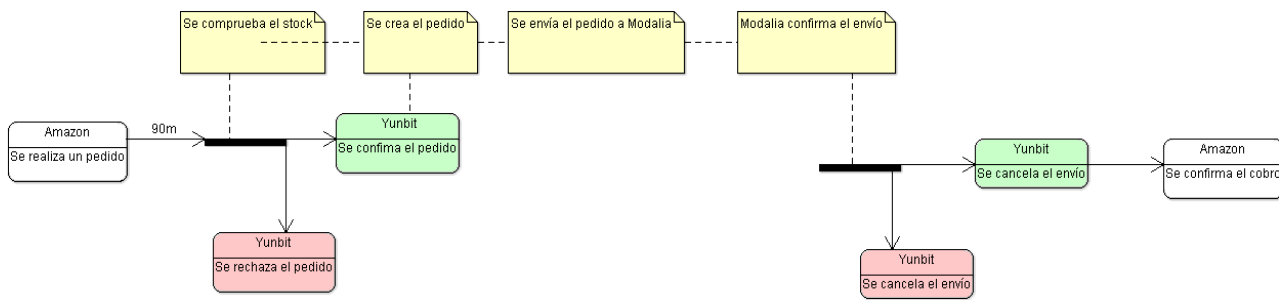


Figure 9: Ciclo de vida de un pedido en el sistema

Este sistema se basa en una tarea que efectúa una llamada a modo de polling cada media hora a los servidores de Amazon, pidiendo los pedidos creados o que hayan cambiado de estado en desde la última llamada. Si comparamos el diseño de nuestro ciclo de vida (Figura 8) con el de amazon, que podemos ver en la sección de análisis (Figura 6), podemos comprobar que respetamos sus puntos de ruptura y su ciclo de vida.

4.2.6 Alertas

Como funcionalidad adicional, la API de Amazon ofrece una variedad de opciones que pueden servir para incrementar la calidad de la experiencia de venta del vendedor y mejorar sus ventas. Para este proyecto se ha implementado sólo una tarea básica de estudios de productos de la competencia, aunque al final de este documento se van a detallar futuras ampliaciones de este proyecto.

4.2.7 Precios

Dado que la API de Amazon MWS permite obtener información de los distintos mercaderes de cada producto, se ha implementado una tarea que diariamente pida a Amazon un informe con los precios que tienen cada vendedor de los productos que hay subidos.

En caso de que haya vendedores que hayan superado en precio a los productos del cliente, tanto como por ofertas o por precio base, se informará con un correo de cada caso (todos en el mismo correo) al cliente, para que pueda actuar para posicionar mejor sus productos, como, por ejemplo, realizando una oferta flash.

5 Implementación

En esta sección se van a exponer los detalles técnicos acerca de cómo se ha programado el proyecto, desgranando cada parte para acabar teniendo una visión más minuciosa acerca del sistema.

5.1 Productos

Como ya se ha comentado, la unidad base de este proyecto es el contenido de los productos. Pero existe la complicación de que varían en cada caso, teniendo tres distintos modelos de datos: los de Amazon, los de Modalia y los de Yunbit.

Para solucionar el problema con Modalia, ya que el receptor de los datos es la empresa Yunbit, se realiza un mapeo de la información, proporcionada mediante unos XMLs en un proceso de sincronización ya existente. En cambio, de cara a Amazon, y dado que se quiere hacer genérico y reutilizable, ha hecho falta implementar un modelo de datos igual que el suyo para que se pueda realizar el intercambio de datos con normalidad, independientemente de la estructura del proyecto, por si cambia el modelo de datos en el siguiente sistema a implementar este módulo. En la *figura 10*, en azul, se puede ver el modelo de datos simplificado de los productos.

5.1.1 Modelo de datos de Amazon Marketplace

Amazon usa un modelo de datos muy particular cuya singularidad consiste en que se puede dividir en diferentes unidades individuales, sostenidas por relaciones de tipo parentesco: **Productos**, **Inventario**, **Imágenes**, **Precios**, **Relaciones**, siendo la unidad más importante a la hora del procesado la de Productos.

En esta memoria, se tratará de simplificar los detalles acerca de los distintos atributos de cada clase, menos los principales que se explicarán más en profundidad, pero primero se destacan varios puntos importantes, tanto de definición como de atributos, que servirán de identificación:

- **SKU**. El SKU es el atributo de identificación única de cada producto, por parte del vendedor, es decir, en este caso por parte de Yunbit, y servirá de relación entre cada producto de Amazon y de Yunbit (Clave primaria).
- **ASIN**. El ASIN es un atributo de identificación única de Amazon. Este atributo se almacena en la base de datos de Yunbit una vez se genere por parte de Amazon, aunque por mero valor informativo, ya que basta el SKU para establecer la relación e identificación.
- **Tipo de mensaje**: Se recogen los distintos tipos: Producto, Inventario, Imágenes, Precio o Relaciones. Cada paquete sólo puede contener mensajes de un mismo tipo.
- **StandardProductID**: Amazon, aparte del SKU y ASIN, utiliza un atributo de identificación genérico, que puede ser un EAN, por ejemplo, utilizado a nivel global en los mercados online para identificación. Si no se dispone de dicho atributo, por ejemplo en el que uno sea el fabricante (como en este caso), se puede solicitar una exención que, una vez conseguida, se pasará a identificar por un ID asignado por AMAZON, el GCID, la marca/fabricante y la ID del producto.

Tras analizar los distintos XSDs, se decidió utilizar un sistema de clases, basadas en interfaces y clases abstractas que recogen los métodos comunes, como son la creación de los XMLs con datos o la validación, para poder construir a partir de ellos los objetos de tipo Amazon, siendo necesario ‘hidratar’ dichos objetos con datos de cada producto, para lo que se ha creado una clase propia para el producto de Alma en pena, que rellena las clases con los datos disponibles, sirviendo de interfaz entre ambos modelos.

En el siguiente diagrama (figura 6) se puede apreciar la organización de clases que el alumno ha implementado, tras analizar los XSDs de Amazon, para la construcción de los productos de tipo Amazon.

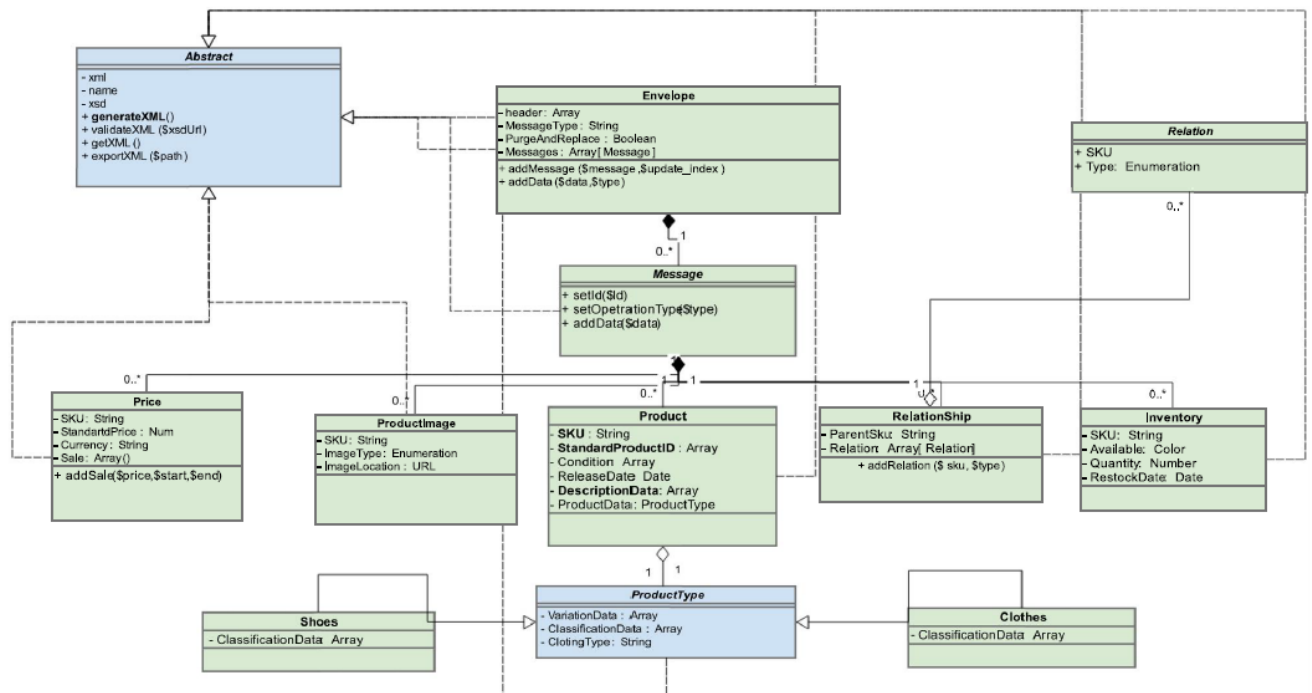


Figura 10: Diagrama de los objetos desarrollados por el alumno

Las funciones a destacar son *validateXML*, que contrasta el XML creado a partir de su objeto con el XSD correspondiente, devolviendo true o false según el resultado, y *generateXML*, que obtiene el XML a enviar a Amazon. Más adelante se explicará cómo funciona el proceso de sincronización.

En estos objetos se han incluido todos los atributos de Amazon, independientemente de si se usan o no, dado que es posible que futuros clientes (o clientes actuales que compren este módulo) puedan necesitarlo.

5.1.2 Modificaciones en los productos

Durante el análisis inicial de los productos de Amazon, se han identificado varios aspectos carentes en los productos de Almaenpena, y otras características y conceptos que se han aprendido para la realización de futuros modelos de datos de productos. Dado que esto último queda fuera del alcance de este proyecto, este trabajo se limitará a ver los cambios realizados en el modelo de datos de Almaenpena (los gestionados por Yunbit) para aumentar la eficiencia de la sincronización con Amazon. Será necesario, primero, describir varios tipos de atributos de Amazon:

- **Bulletpoints.** Estos atributos son listados que permiten dar una descripción personal de cada producto (padre y/o variación). Están limitados a un máximo de 5 por producto.
- **Keywords.** Como se indica, supondrá una lista de términos separados por comas para ayudar en el posicionamiento del producto.
- **Category.** Este es, a diferencia de los otros listados arriba, un atributo obligatorio por producto. Determinará a qué sección del amplio catálogo de Amazon pertenece, pudiéndosele asignar como mínimo 1 sección, y 2 como máximo. Un ejemplo, por ejemplo, para un zapato de tacón, sería *Zapatos > Zapatos mujer > Zapatos de tacón*. Cada categoría se distingue como un árbol, siendo recomendable incluir los productos en nodos hojas.

En la figura 11 se identifican todas las categorías utilizadas por Amazon y su respectivo código:

Árbol de navegación de categorías de Amazon

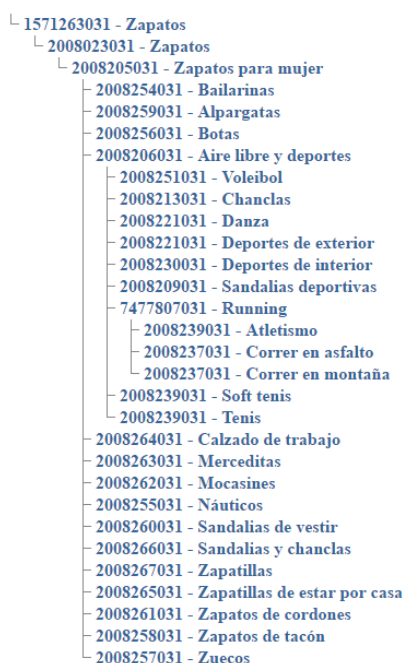


Figura 11: Árbol de categorías de Amazon creado por el alumno

Así mismo, es necesario guardar los atributos de Amazon, como son el código ASIN, por lo que fue necesario ampliar el modelo de datos y los gestores para guardar y poder acceder a estos datos.

5.2 Sincronización

Una de las claves de este proyecto consiste en la sincronización entre distintos servicios, ya que de otra manera no tiene sentido. La idea es que el gestor de Alma en pena de la aplicación no tenga que hacer nada y sea la aplicación la que, de forma autónoma, se encargue de mantener los listings sincronizados. Así, se distinguen tres tipos de sincronización: de productos con Modalia, de productos con Amazon, y de pedidos.

5.2.1 Modalia

Modalia es el gestor de productos que usaba antes la empresa Alma en pena, pero, aunque la idea es, en un futuro cercano, dejar de usarlo y usar únicamente los servicios de Yunbit, aún se inserta contenido a través de Modalia, por lo que hace falta un medio para poder sincronizarse con Modalia y recoger los contenidos que hayan actualizado los vendedores desde el gestor de Modalia.

Para esto, existe una tarea que todos los días, extrae el catálogo de productos activos de Modalia, en formato XML, y lo sube a yunbit, actualizando todos los datos. Aprovechando este proyecto, dicha tarea se ha modificado para aumentar su eficiencia, de manera que ahora compara el contenido que hay en la plataforma con el de Modalia, y sólo actualiza en caso de que haya alguna novedad o modificación (por ejemplo, una imagen se haya modificado, o un texto descriptivo, o el stock).

Al modificarlo, comprobará si este producto está presente en algún listing de productos para Amazon. De ser así, lo marcará como no sincronizando, haciendo de esta manera que se actualice en la tarea de sincronización con Amazon diaria. Este cambio en el estado de sincronización, también se realizará cuando se realice algún pedido o se modifique desde el gestor, ya que se ha realizado mediante triggers de inserción, modificación y eliminación de la base de datos.

5.2.2 Amazon MWS

Este subsistema incluye el proceso completo de subida de productos a Amazon. Esta subida se ha realizado de dos modos: a) mediante la creación de una tarea de activación manual y b) mediante una tarea programada que cada día revise el estado de los listings y productos de los mismos, subiendo nuevos listings, desactivando listings inactivos, actualizando stock y productos modificados y, en caso de que se hayan borrado productos de los listings, borrando los productos de Amazon.

Para esto, se ha desarrollado una clase Cliente genérica de Amazon, en PHP, que gestiona todas las comunicaciones con la API de Amazon Marketplace. Todo este proceso se realiza mediante mensajes del tipo **SubmitFeed** a Amazon, que, a su vez, se distinguen

los tipos ya indicados: *Productos*, *Inventario*, *Precios*, *Imágenes* y *Relaciones*. Un paquete de SubmitFeed sólo puede contener mensajes de un único tipo.

Para el borrado de productos, se ha implementado es un trigger en la BD que, tras eliminar un producto de la lista de productos de un listing, comprueba la existencia de este producto en otros listings activos. Si no existe, se añadirá a una cola de productos a borrar. Así mismo, se ha creado otro trigger al insertar un producto que saca de la cola ese producto en caso de estar presente, para evitar que se borre.

Tras el intercambio de paquetes con Amazon, se esperará al procesado de datos por parte de amazon y se le pedirá un informe de estado de los productos enviados en el paquete de datos XML. Con ese informe, tras sincronizar todos los listings, se actualizará el estado de sincronización de los listings y de los productos, estando los estados de ‘Pendiente de sincronizar’, ‘Sincronizado’, ‘Con problemas’ y ‘Con errores’, y guardará el log de problemas y errores, para poder identificarlos y corregirlos. También se enviará un email al cliente y al soporte de Yunbit con el estado de la sincronización.

5.3 Pedidos

Al igual que con los productos, hay una confusión entre los pedidos ya que existen tres tipos de pedidos, por lo que ha hecho falta sincronizar si se quiere proporcionar un servicio fiable, y no llegar a problemas con, por ejemplo, falta de stock, lo cual empeora la calidad del servicio a ojos del consumidor.

Dado que no se puede controlar de forma unificada, ya que cada uno de los sistemas (Yunbit, Amazon, Modalia) realiza sus gestiones de pedido de diversa forma, lo que sí que se puede hacer es llevar la cuantía de stock en los servidores de Yunbit, centralizando la sincronía de pedidos y realizando desde ahí la validación de pedidos. Con unos buenos márgenes de tiempo entre sincronizaciones, se puede evitar el caso de tener que cancelar pedidos, lo que perjudicaría la puntuación de Alma en pena en Amazon.

Para esto, se realiza una sincronización de los pedidos diaria con Modalia, ya que, al gestionar las tiendas físicas, no tienen una regularidad de actualización definida. Esta sincronización crea los pedidos realizados por modalia en la base de datos de Yunbit.

5.3.1 Pedidos de Amazon

Durante el análisis y el diseño se ha visto el funcionamiento de los pedidos: un cliente de Amazon realiza un pedido y entra en el ciclo de vida, con sus puntos de ruptura, antes de que el cliente realice el pedido.

Como ya se ha mencionado durante el diseño, se ha utilizado un sistema de *polling* con Amazon para llevar un seguimiento de los pedidos. Mediante un mensaje de tipo **ListProducts**, se solicitan todos los pedidos con fecha de creación o actualización posterior a la última llamada de revisión de estado de pedidos.

Cuando se reciban pedidos como respuesta de dicha llamada, se procederá a crear un usuario en el sistema Yunbit (si no existe ya un usuario dado de alta con ese correo, en

tal caso usará ese) y guardará los datos de la dirección y del pedido. Si todos los productos del pedido se encuentran disponibles, envía el mensaje de confirmación del pedido. Si, en cambio, el pedido ya existía y es una cancelación, se cambiará el estado del pedido y se comunicará a Modalia, siempre y cuando aún no se haya realizado el envío. Como detalle para el consumidor, se da un espacio de una ejecución de la tarea (una hora) para que se pueda cancelar el pedido antes de tramitar el envío a Modalia.

Posteriormente, envía a Modalia la información sobre el nuevo pedido o la cancelación del pedido existente. Si es la creación del pedido, informará a Yunbit cuando se procese para el envío, momento en el cual el sistema informará a Amazon del inicio del proceso de envío, momento en el cual el comprador dejará de poder cancelar el pedido.

Para aceptar o rechazar un pedido entero (por falta de stock, o por problemas de envío reportados por Modalia) se usa un mensaje de tipo **AcknowledgeOrder** con el estado como suceso o error. Una vez que se cancela un pedido, se le devuelve al comprador todo el dinero y se deja de ingresar.

Si se desea cancelar sólo un producto del pedido, por ejemplo, que en un producto de los dos del pedido no haya stock, pero en el otro sí, en vez de cancelar el pedido, se cancela el producto del pedido mediante un mensaje de tipo **OrderAdjustment**.

Para indicar, cuando Modalia confirme el envío del producto, que este se ha producido, se envía un mensaje de tipo **OrderFulfillment** marcando la fecha de envío y la fecha estimada de entrega. Así mismo, se indica el código de seguimiento ofrecido por la empresa de logística. Es en este momento, a partir de que se indica que se ha producido el envío, cuando Amazon da el dinero de la compra al vendedor.

5.3.2 Actualización de stock

Cada vez que se sincroniza la información de stock desde Modalia (con la tarea diaria) o se hace un pedido desde el Ecommerce de Yunbit, hay un riesgo de que se produzca un desfase en el stock de los productos.

Para evitar esto, se ha creado una cola de actualización de stock. Cada vez que cambie el stock con un pedido o tras sincronizar los datos con Modalia, si hay un cambio de stock se añadirá el Id del producto a una cola. Se ha diseñado una tarea automática que revise dicha cola y, en el caso de haber elementos, envíe un mensaje de actualización de Inventario a Amazon. Esto se ha hecho así para no saturar a Amazon en el caso de que haya, por ejemplo, varios pedidos seguidos. Así sólo manda un máximo de dos mensajes a la hora.

5.4 Interfaz del usuario

La interfaz del usuario está implementada mediante el framework de Yunbit, GeaWebblocks, que permite, de manera sencilla, implementar listados y formularios para gestionar los datos. Para varias funcionalidades que no incluía ha sido necesario cierto grado de personalización, pero teniendo como criterio mantener la coherencia con el resto de la interfaz gráfica de la aplicación de Yunbit.

A continuación podemos ver una breve descripción de estas interfaces

5.4.1 Gestores de Listings

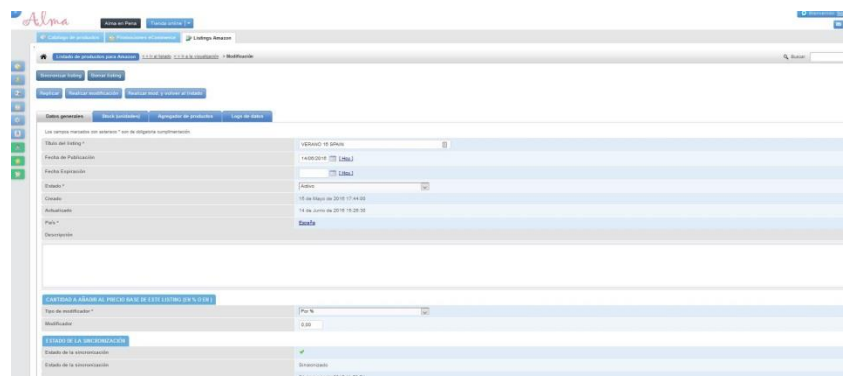
Como se ha mencionado durante todo el documento, la unidad de organización de los productos a subir a Amazon son los listings. El cliente puede matener varios listings activos que se sincronizarán con Amazon, y podrá desactivarlos, modificarlos o editarlos desde el propio gestor de Yunbit.



Título del listing	País	Estado	Creado	Actualizado	Fecha de Publicación	Fecha Expiración	Tipo de modificador	Modificador	Estado de la sincronización	Estado de la sincronización
VERANO 18	España	Inactivo	28 de Abril de 2018 09:59:00	28 de Abril de 2018 10:00:12			Por %	0.00	✓	Sincronizado
VERANO 18	Argentina	Inactivo	28 de Abril de 2018 09:59:00	28 de Abril de 2018 10:01:25			Por %	0.00	⚠	No sincronizado
VERANO 18 SPAIN	España	Activo	19 de Mayo de 2018 17:44:00	14 de Junio de 2018 18:28:30	14 de Junio de 2018		Por %	0.00	✓	Sincronizado
Test Subida Amazon	España	Inactivo	13 de Junio de 2018 18:27:15	13 de Junio de 2018 18:27:15			Por %	100.00	⚠	No sincronizado
VERANO 18 Spain	España	Inactivo	21 de Junio de 2018 13:05:28	21 de Junio de 2018 13:05:28			Por %	0.00	⚠	No sincronizado

Figura 12: Listado de listings

Dentro del gestor, podemos acceder a tres pestañas. La primera, que se puede ver arriba, es la configuración de los datos generales, y donde se muestra el estado de la sincronización. Podemos apreciar varios estados de sincronización: Sincronizado, Con Errores, Con Alertas, Error, No sincronizado. En los tres primeros, los productos se habrán podido subir correctamente a Amazon, pero en el 2º y 3º habrá ocurrido algún error o alerta de algún producto al procesar Amazon los datos, y el gestor deberá revisarlo.



Lista del listing	VERANO 18 SPAIN
Fecha de Publicación	14 de Junio de 2018 18:28:30
Fecha Expiración	14 de Junio de 2018
Estado	Activo
Creado	19 de Mayo de 2018 17:44:00
Actualizado	14 de Junio de 2018 18:28:30
País	España
Descripción	
Tipos de modificador	Por %
Modificador	0.00
Estado de la sincronización	✓
Estado de la sincronización	Sincronizado
Estado de la sincronización	21 de Junio de 2018 13:05:28

Figura 13: Datos generales del gestor de Listings

Para poder ver los productos que contiene cada Listing, tenemos la segunda pestaña. En ella podemos ver un listado con todos los productos y su estado individual de sincronización, los cuales son: Sincronizado, Error, Alerta, No sincronizado. Desde cada file podemos tener acceso a la ficha del producto, que veremos más adelante. Así mismo, al entrar en detalle en un producto con algún tipo de error, veremos los errores que ha devuelto Amazon para poder corregirlos.

Inicio

Stock productos

Agregar un producto

Logueo Usuarios

Los campos marcados con asterisco * son de obligatoria cumplimentación.

Stock productos50

Productos

Lista de productos para Anunciar

+ Listado

Filtro

Buscar

Actualizar

Borrar selección

1 - 20 (de un total de 16 filas - 2 páginas) |

1

2

30

por página

	Actualizado	Producto *	Categoría	Etiqueta	Estado *	Última sincronización	Estado de la sincronización
<input type="checkbox"/>	✓	16 de Junio de 2018 01:40:37	V1001 - 2018-06-01-11-47	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Junio de 2018 01:40:37	V1002 - 2018-06-01-11-48	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Junio de 2018 01:40:37	V1003 - 2018-06-01-11-50	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:58:16	V1004 - 2018-05-16-11-10	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:58:16	V1005 - 2018-05-16-11-10	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1006 - 2018-05-16-11-10	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1007 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1008 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1009 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1010 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1011 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1012 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1013 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1014 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1015 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1016 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1017 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1018 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1019 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1020 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1021 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1022 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1023 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1024 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1025 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1026 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1027 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1028 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1029 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1030 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1031 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1032 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1033 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1034 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1035 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1036 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1037 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1038 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1039 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1040 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1041 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1042 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1043 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1044 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1045 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1046 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1047 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1048 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1049 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1050 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1051 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1052 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1053 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1054 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1055 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1056 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1057 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1058 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1059 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1060 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1061 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1062 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1063 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1064 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1065 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1066 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1067 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1068 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1069 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1070 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1071 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1072 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1073 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1074 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1075 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1076 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1077 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1078 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1079 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1080 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1081 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1082 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1083 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1084 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1085 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1086 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1087 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1088 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1089 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1090 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1091 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1092 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1093 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1094 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1095 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1096 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1097 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1098 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	V1099 - 2018-05-16-11-09	Botones	Activo	21 de Junio de 2018 11:23:24	Sincronizado
<input type="checkbox"/>	✓	16 de Mayo de 2018 08:57:19	<				

Para añadir productos, tenemos una tercera pestaña, en la que se pueden seleccionar, desde un listado con varias opciones de filtros, todos los productos que se deseen, y con el botón de Agregar productos seleccionados, se añaden al listing. En la Figura 15 podemos ver dicho agregador.

Datos generales | **Stock (Unidades)** | **Agregar de productos** | **Logs de datos**

Los campos marcados con asterisco * son de obligatoria cumplimentación.

Agregar de productos

+ Listado

Agregar productos seleccionados

Filtro:

Categoría: TODOS Estado: TODOS Etiquetas: EDCO, BOMBA, BATERIAS, Display plano Fecha de creación: Desde Hasta

Aplicar Filtros Eliminar Filtros

1 - 33 (de un total de 533 filas / 22 páginas) | Listando 30 por página

	Fecha de creación	Modelo *	Categoría	Estado
<input type="checkbox"/>	16 de Septiembre de 2016 01:11:00	1166	00015	Activo
<input type="checkbox"/>	4 de Febrero de 2016 01:07:00	50621	VERI10	Activo
<input type="checkbox"/>	26 de Noviembre de 2014 01:12:26	CL1	00014	Inactivo
<input type="checkbox"/>	1 de Noviembre de 2014 01:07:00	114300	00014	Inactivo
<input type="checkbox"/>	1 de Noviembre de 2014 01:07:05	114301	00014	Inactivo
<input type="checkbox"/>	1 de Noviembre de 2014 01:07:05	114302	00014	Inactivo
<input type="checkbox"/>	1 de Noviembre de 2014 01:07:05	114304	00014	Inactivo

Figura 15: Agregador de productos

Por último, aunque sólo visible para los administradores, hay una 4º pestaña, que guarda los logs de sincronización con Amazon: El estado de cada sincronización (de Productos, Stock, Imágenes...) y las peticiones enviadas a Amazon, su respuesta, y su informe de procesado de datos

Lógica de negocio						
Las variables manipuladas con esta lógica se ven de siguiente forma:						
<div> <div>Lógica de negocio</div> <div> <div>Ver</div> <div>Editar</div> <div>Eliminar</div> </div> </div>						
<div> <div>Ver</div> <div>Editar</div> <div>Eliminar</div> </div>						
Estado	Nombre del negocio	Tipo de operación	Categoría	Clase de la implementación	Estado	Creación
<input checked="" type="checkbox"/>	VERIFICAR (S. 2016)	Subnet-Feed	PRODIGIO	STG07364761	En proceso	21 de junio de 2016 11:21:19
<input checked="" type="checkbox"/>	VERIFICAR (S. 2016)	Subnet-Feed	MANEJ	STG07364767	Finalizada	21 de junio de 2016 11:42:21
<input checked="" type="checkbox"/>	VERIFICAR (S. 2016)	Subnet-Feed	MANEJ	STG07364767	Finalizada	21 de junio de 2016 11:42:21
<input checked="" type="checkbox"/>	VERIFICAR (S. 2016)	Subnet-Feed	MANEJ	STG07364767	Finalizada	21 de junio de 2016 11:42:21

Figura 16: Listado de logs de subida

5.4.2 Gestores de productos

Como ya se ha dicho previamente, los productos y el gestor ya estaban desarrollados, pero se les han hecho modificaciones para adaptarlos al modelo de datos de Amazon. Lo que se ha hecho es añadir dos nuevas opciones de datos: Los Bulletpoints, o puntos de listado, y las características de los productos.

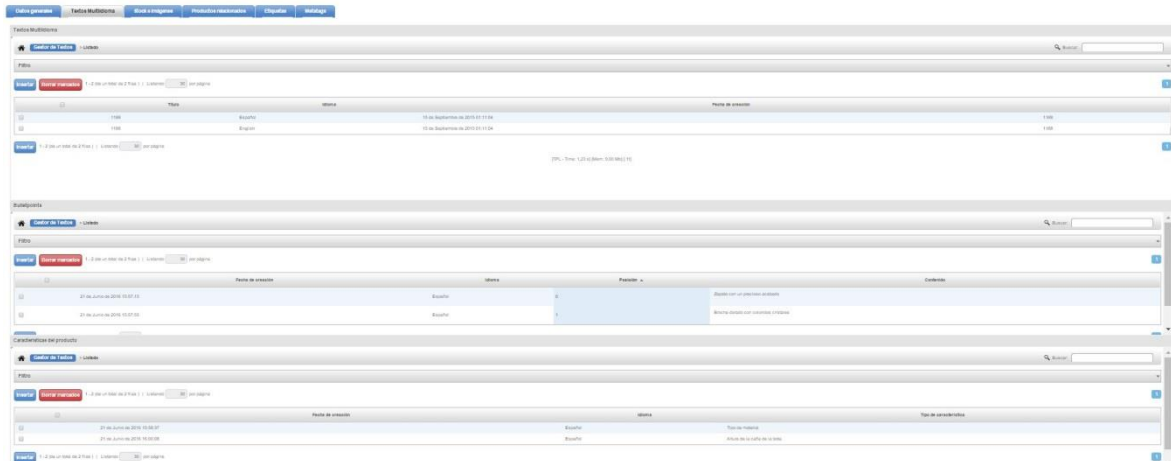


Figura 17: Gestor de características y bulletpoints de un producto

A continuación se puede ver el proceso de inserción de una nueva característica de Amazon. Se ha hecho por dos pasos, al dar la opción de multiidioma, es decir, de tener el contenido tanto en español como inglés, dado que Alma en pena vende sus productos también fuera de España.

Los campos marcados con asterisco * son de obligatoria cumplimentación.

Página web de relación	cw55f754087bada
Fecha de creación	21 de Junio de 2016 15:58:37
Idioma *	Español
Tipo de característica *	Tipo de material
Contenido *	Cuero y tela

PREINSERCIÓN

Introduzca los datos para el nuevo registro Gestor de Textos

Tipo de característica: Tipo de material

Idioma: Español

Aceptar Cancelar

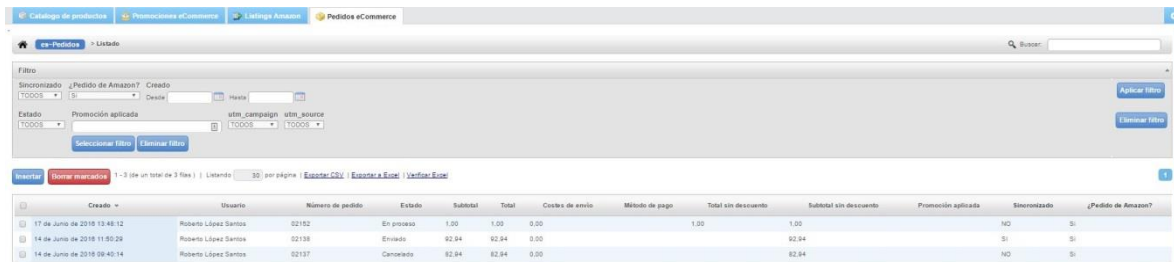
Replicar Terminar inserción

- Algodón
- Ante
- Caucho
- Charol
- Cocodrilo
- Cuero
- Cuero nobuk
- Cuero y goma
- Cuero y rejilla
- Cuero y sintético
- Cuero y tela**

Como podemos apreciar en la segunda imagen, en se da una lista de valores cerrados, en un desplegable, para esta característica. Cada tipo de característica está personalizada, con sus respectivos valores de Amazon y, en caso de las tallas, permitiendo la inserción de números.

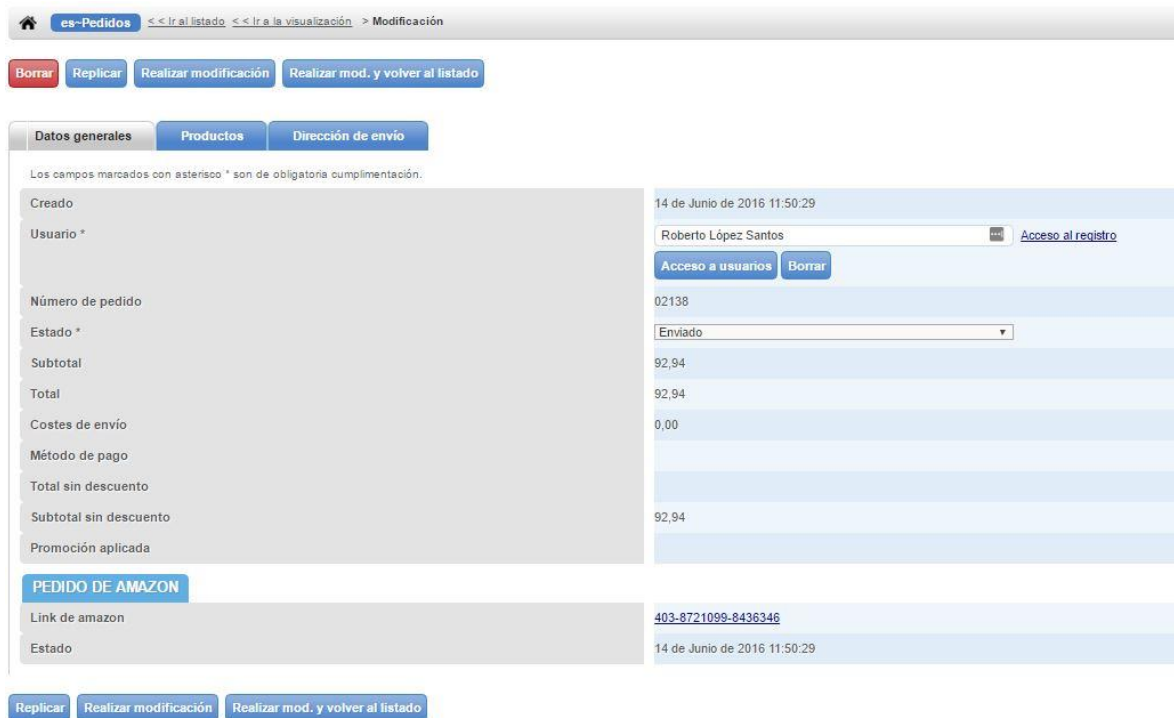
5.4.3 Gestores de pedidos

Los pedidos ya estaban en la plataforma de Yunbit, pero se han modificado para poder diferenciar los pedidos de Amazon de los de la tienda online. Se ha añadido en el listado un nuevo filtro, para determinar si filtrar por pedidos de Amazon o web, y, si el pedido es de Amazon, dentro del pedido podemos ver el estado del pedido en Amazon y un link que nos dirige al gestor de ese pedido de Amazon Marketplace, para realizar funcionalidades que no contemplamos (en este momento) como es el reembolso de los pedidos.



	Creado	Usuario	Número de pedido	Estado	Subtotal	Total	Costes de envío	Método de pago	Total sin descuento	Subtotal sin descuento	Promoción aplicada	Sincronizado	¿Pedido de Amazon?
17 de junio de 2016 13:45:12	Roberto López Santos	02192	En proceso	1,00	1,00	0,00		1,00	1,00		NO	SI	
14 de junio de 2016 11:50:29	Roberto López Santos	02138	Enviado	92,94	92,94	0,00			92,94		SI	SI	
14 de junio de 2016 08:40:14	Roberto López Santos	02137	Cancelado	92,94	92,94	0,00				92,94	NO	SI	

Figura 18: Listado de pedidos



es-Pedidos << Ir al listado << Ir a la visualización > Modificación

Borrar Replicar Realizar modificación Realizar mod. y volver al listado

Datos generales Productos Dirección de envío

Los campos marcados con asterisco * son de obligatoria cumplimentación.

Creado	14 de Junio de 2016 11:50:29
Usuario *	Roberto López Santos Acceso al registro
Número de pedido	02138
Estado *	Enviado
Subtotal	92,94
Total	92,94
Costes de envío	0,00
Método de pago	
Total sin descuento	
Subtotal sin descuento	92,94
Promoción aplicada	

PEDIDO DE AMAZON

Link de amazon	403-8721099-8436346
Estado	14 de Junio de 2016 11:50:29

Replicar Realizar modificación Realizar mod. y volver al listado

Figura 19: Gestor de pedidos

5.5 *Alerta de precios*

Para realizar el análisis de precios, el sistema Yunbit pedirá a Amazon información sobre los productos que hayan sido subidos por los demás vendedores que compartan Id con los nuestros. De esta manera, Amazon devolverá todos los precios de estos productos, tanto los rebajados por promociones activas, como los normales.

Con estos datos, el sistema elaborará un informe avisando a los administradores si el precio de algún producto no es el más bajo del mercado y, si ha sido superado, si se trata de una oferta o no.

De esta manera, el administrador podrá tomar la decisión de si mantener los precios, bajarlos, o crear una oferta, para facilitar que su producto se venda frente al resto de mercaderes de Amazon.

6 Pruebas y validación

El objetivo de las pruebas es detectar y solventar los fallos producidos por el sistema para comprobar la correcta funcionalidad del sistema, así como realizar una validación de las funcionalidades y requisitos listados durante el análisis.

6.1 Estrategia de pruebas

Dada la variedad del sistema desarrollado, ya que consta de diversas partes diferenciadas (Productos, Pedidos, Conexiones con los servidores, Tareas automáticas, etc.), se ha seguido un modelo incremental de testeo.

Este modelo parte de unas pruebas unitarias para cada módulo, y se han usado dos herramientas de automatización de pruebas para este proceso: PHPUnit y Selenium (usado para validar varios procesos que requerían de la interfaz gráfica. Por problemas de compatibilidad con PHP se ha usado Selenium sobre Django).

Tras esto, se ha usado un proceso bottom-down diferenciado para integrar el proyecto. Esto quiere decir que se ha dividido el proyecto en dos partes diferenciadas, Pedidos y Productos, y se ha ido integrando cada parte desde la creación de los mensajes hasta el establecimiento de las comunicaciones entre los distintos servidores.

Por último, ya que es un proyecto para un cliente, se han hecho varias pruebas de usabilidad antes de presentárselo a Alma en pena. Estas pruebas han sido realizadas por tres personas, el alumno, Roberto López y un desarrollador de Yunbit.

Por motivos de falta de espacio, se incluirán volcados de datos y archivos XML en los anexos.

6.2 Desarrollo de las pruebas

6.2.1 Pruebas unitarias

Como se ha dicho, se han realizado pruebas unitarias automatizadas mediante dos herramientas: PHPUnit y Selenium. PHPUnit permite automatizar pruebas de lógica, y Selenium es una herramienta de testeo de front end que permite realizar macros de acciones y valorar los resultados obtenidos, por lo que se ha usado para realizar las pruebas con los gestores y los resultados de los cambios de estado.

Para probar los productos, primero se realizó una integración del módulo de creación de objetos del modelo de Amazon con el de Alma en pena, y se crearon los XML y se validaron a través de los XSD de Amazon. Para probar los envíos de datos a Alma en pena, se utilizaron Mocks, respuestas XML precreadas que da amazon, y se interpretaron como si fueran reales. Al mismo tiempo, se probaron que los distintos procesos y triggers cambiaran el estado de la base de datos para marcar los productos como actualizados al realizar cambios o hacer pedidos desde la web.

Con los pedidos, se hizo lo mismo que con los productos. También se usaron mocks de prueba y se validó la creación de los distintos mensajes de cambios de estado

para Amazon, rellenando toda la casuística: pedidos correctos, cancelación por parte del usuario, por parte de Modalia, falta de stock y falta de stock de sólo parte del pedido.

6.2.2 Pruebas de integración

Durante la integración, se fue montando el proyecto de forma que se adaptara la personalización de los productos a las conexiones con Amazon, y se unieran las dos partes.

Lo primero que se hizo para la integración, al verse necesario, fue hacer un sistema de logs que recogiera las peticiones realizadas y mensajes recibidos de Amazon y los guardara, para poder validar su correcto contenido. Esto servirá en un futuro para comprobar problemas que puedan ocurrir durante en el funcionamiento de la aplicación.

Tras eso, se juntó la creación de paquetes XML con la conexión con Amazon, y se comprobó que los listings se enviaran correctamente a Amazon y este los creara correctamente en el gestor. Esta fue la parte más compleja, debido a la mala calidad de la documentación y al escaso y pésimo soporte técnico que ofrece Amazon.

Con los pedidos, lo que se ha hecho es crear un pedido con la cuenta del alumno, y realizar la casuística con ese pedido. Se ha dejado la confirmación de envío para el final, para no cobrar al alumno los pedidos, aunque se puede realizar el reembolso. Con ello, se ha probado a cancelar el pedido por parte del usuario, a cancelar el pedido al recibirlo, a pedir más stock del que hay (cambiando las cantidades de Amazon a mano), y otros casos, validando al final con Modalia que se produzca la correcta creación del paquete y orden de distribución.

Tras tener cada parte funcionando, se han probado las tareas programadas mediante llamadas manuales, sin meterlas en el crontab, y sólo una vez que se ha validado su funcionamiento deseado, se han incluido en el crontab de tareas del servidor.

6.2.3 Pruebas de usabilidad

Para las pruebas de usabilidad, se han usado a personas ajenas al desarrollo técnico del proyecto y familiarizadas con el framework GeaWebBlocks de Yunbit. Se han puesto distintos casos de uso de la aplicación y se ha comprobado si es fácil realizar las acciones desde la web.

Se ha visto que hay limitaciones en el proyecto, como, por ejemplo, devolver un pedido, pero estas limitaciones no entraban en el alcance inicial, por lo que se han propuesto como ampliaciones, lo cual se verá al final de esta memoria. Exceptuando estos comentarios menores, la interfaz ha sido muy bien valorada por los usuarios, que han podido realizar de manera sencilla las tareas oportunas.

6.3 Validación

La validación se ha dividido en tres fases: Una primera fase realizada por el alumno, donde ha valorado el alcance de lo desarrollado y si ha cumplido con los objetivos, después, junto a las pruebas de usabilidad, por parte de su jefe, Roberto López, se ha producido una validación del alcance original, y por último, la tercera fase se ha producido durante la formación de los clientes en el uso del módulo.

Durante la primera validación, se ha puesto a prueba el sistema y se ha comprobado que ha cumplido con los requisitos del proyecto y ha llegado al alcance como para poder proceder a las siguientes fases de validación:

- Requisitos Funcionales cumplimentados: 19/19
- Requisitos No Funcionales cumplimentados: 10/11

El RNF que no se ha cumplimentado, es el RNF09: Crear mensajes de errores accesibles para los usuarios. En un futuro próximo se hará un diccionario que traduzca los errores de Amazon en mensajes accesibles.

Tras estos resultados, se ha procedido a la validación por parte del tutor, Roberto López, del resultado final, y las pruebas de usabilidad realizadas por un desarrollador de la empresa externo al desarrollo de este proyecto. Ambos han validado la funcionalidad del proyecto con éxito.

Por último, el jueves 16 de Junio de 2016, se realizó una formación con el cliente de Alma en pena, donde se presentó el proyecto desarrollado y se les instruyó en el manejo del mismo. De dicha formación se sacaron diversas posibles mejoras, que no entraban en el alcance del proyecto, como pueden ser las devoluciones de productos, y que se tendrán en cuenta para un futuro.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

En este trabajo de fin de grado se ha realizado un proceso de Ingeniería de Software junto con un proceso de gestión de proyectos, al ser el alumno el que ha trazado el alcance, analizado y diseñado el proyecto y ha tenido las reuniones con el cliente, a la vez de encargarse él mismo del desarrollo.

Este proyecto también supone un gran avance para el producto de Yunbit, ya que a partir de ahora puede ofrecer a los comercios electrónicos que soporta y a nuevos clientes este módulo, que le permitirá vender los productos a través de Amazon, el mayor portal de ventas de productos online, dando a la marca, además de mayor alcance, una mayor visibilidad y promoción.

Al inicio del proyecto, el estudiante ya disponía de una base de conocimientos amplia, pero le ha proporcionado experiencia, entre otros aspectos, en el uso de APIs web, como es la API de Amazon, y ha ampliado su conocimiento en conexiones con servidores de tercero y de modelado de datos. Así mismo, ha profundizado su conocimiento en desarrollo web y en el uso de automatizaciones de pruebas con Selenium, herramienta muy útil como ayuda en la fase de pruebas. Por último, cabe desatacar que el hecho de analizarlo a la vez como un proyecto de uso genérico y particular, ya que el propósito era ser usable por Alma en pena pero también por futuros proyectos, ha hecho que sea esencial al uso de modulación y parametrización para este proyecto, lo que ha resultado complejo como se ha demostrado a lo largo de este trabajo pero se ha concluido con éxito.

7.2 Trabajo futuro

Tras concluir el desarrollo, se ha obtenido una primera versión de la integración con Amazon. Al ser un proyecto muy interesante para vender a los clientes, se le dedicarán en el futuro más recursos para su ampliación y las líneas a ampliar serán las siguientes:

- Ampliar la gestión de pedidos por parte de Yunbit para poder devolver el dinero al comprador al cancelar un pedido directamente desde la plataforma.
- Hacer un módulo de devoluciones de producto e integrarlo con Amazon.
- Gestionar desde la plataforma de Yunbit la gestión de precios de envío.
- Ampliación a distintos Marketplaces europeos: Francia, Alemania, Inglaterra, Portugal.
- Nuevos tipos de informes para el sistema de Amazon, como un reporte de la satisfacción de los consumidores o un feedback de los comentarios hechos en Amazon por los usuarios, para poder mejorar el servicio.

8 Glosario

- **Back-end**: Es el lado del servidor. Conlleva toda la lógica de procesamiento de datos, de servicios y de base de datos.
- **Bulletpoint**: En el lenguaje de Amazon, son los puntos descriptivos que aparecen en los resúmenes de los productos.
- **Businesscloud**: Es un framework de PHP creado y usado por Yunbit, basado en un modelo MVC.
- **Crontab**: En los sistemas Unix, es el Sistema que se encarga de programar las tareas periódicas del OS.
- **CURL**: Es una librería de funciones usadas para la comunicación entre varios servidores.
- **DDoS**: Denegación de servicio. Dícese de cuando se satura un servidor para impedir su comportamiento normal.
- **E-commerce**: Comercio electrónico.
- **Envelope**: Envoltorio. Es como se conoce en la API de Amazon a los contenedores XML del resto de mensajes, que guardan metadatos sobre el vendedor y el servicio.
- **Framework**: Es un conjunto de herramientas en un determinado lenguaje de programación que forman un entorno de trabajo que facilita las labores de desarrollo.
- **Front-end**: Es la parte visual de una aplicación web, que comprende todo lo que ve el usuario.
- **GeaWebBlocks**: Un framework de modelado de gestores de datos escrito en PHP por la empresa Yunbit.
- **JUnit**: Es un framework de pruebas unitarias para PHP.
- **Listing**: Dentro de las definiciones de la API de Amazon, comprende un determinado conjunto de productos para el Mercado de un país (de amazon.es, en este caso).
- **Market**: Dentro de las definiciones de Amazon, es el espacio de ventas de un país, por ejemplo, amazon.fr para Francia, o amazon.es para España.
- **Mercader**: Es la empresa o persona que pone a la venta productos en Amazon.
- **Mock objects**: Son clases de pruebas, con entradas ya definidas, para realizar pruebas unitarias o de integración.
- **MVC**: Modelo-Vista-Controlador. Es el nombre de una arquitectura de software que separa la lógica de la estructura de datos y de la línea de negocio.
- **Responsive**: Es un diseño que se adapta a distintos dispositivos y resoluciones, además de ser amigable con el usuario.
- **REST**: Representational State Transfer, es un estilo de arquitectura usado para la comunicación con APIs o sistemas web.
- **Selenium**: Es un módulo de pruebas de software, creado para Java pero con integración en .net y python, usado para realizar pruebas y macros que incluyan interacción con un front-end.
- **Stock**: Es el inventario (cantidad de productos) de una tienda.
- **Bandwidth Throttling**: Es un método de regulación de banda para servicios web, que tiene un cupo al tiempo (por ejemplo, 30 mensajes a la hora) y se va recuperando exponencialmente.
- **Variaciones**: En el lenguaje de Amazon, son las variantes de un producto, tales como colores o tallas.

9 Referencias

1. Datos obtenido de sitio web de w3techs:
<http://w3techs.com/technologies/details/pl-php/all/all>
2. Documentación del RFC de PHP: <https://wiki.php.net/rfc/php6>
3. Ranking de db oficial: <http://db-engines.com/en/ranking>
4. Documentación IT de techtarget:
<http://searchitchannel.techtarget.com/feature/MySQL-version-numbers>
5. Wiki de geanet [Link privado]
6. Amazon para desarrolladores : <https://developer.amazonservices.es/>
7. Apartado de FAQs de Amazon :
<https://developer.amazonservices.es/gp/mws/faq.html/276-2002093-9824012>
8. Foro de desarrolladores de Amazon :
<https://sellercentral.amazon.es/forums/thread.jspa?threadID=77084>

10 Anexos

A Ejemplo de comunicación de Productos con Amazon

A.1 Mensaje de producto

```
<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>Product</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <Product>
      <SKU>cw55ef6bb5370f5A</SKU>
      <StandardProductID>
        <Type>GCID</Type>
        <Value>cw55ef6bb5370f5</Value>
      </StandardProductID>
      <ItemPackageQuantity>1</ItemPackageQuantity>
      <NumberOfItems>1</NumberOfItems>
      <DescriptionData>
        <Title> Alma en pena Mujer - I15011</Title>
        <Brand> Alma en pena </Brand>
        <Designer> Alma en pena </Designer>
      </DescriptionData>
      <ProductData>
        <Shoes>
          <ClothingType>Shoes</ClothingType>
          <VariationData>
            <Parentage>parent</Parentage>
            <VariationTheme>SizeColor</VariationTheme>
          </VariationData>
          <ClassificationData>
            <ColorMap>BLACK</ColorMap>
            <SizeMap>36</SizeMap>
            <CountryProducedIn>ES</CountryProducedIn>
            <Department>Mujer</Department>
            <CountryAsLabeled>ES</CountryAsLabeled>
            <ModelName>I15011</ModelName>
            <ModelYear>2015</ModelYear>
            <OuterMaterialType>Leather</OuterMaterialType>
            <CollectionName>Otoño-Invierno</CollectionName>
          </ClassificationData>
        </Shoes>
      </ProductData>
    </Message>
  </AmazonEnvelope>
```

```
</ProductData>
</Product>
</Message>
</AmazonEnvelope>
```

A.2 Mensaje de imágenes

```
<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>ProductImage</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <ProductImage>
      <SKU>cw56c1ddfe72660</SKU>
      <ImageType>Main</ImageType>
      <ImageLocation>https://prod-plat-
almaenpena.geanetondemand.com//EPORTAL_IMGS/GENERAL/ALMAENPENA/IMG
2-ap56c1ddfe75a04/20830-1.jpg</ImageLocation>
    </ProductImage>
  </Message>
  <Message>
    <MessageID>2</MessageID>
    <ProductImage>
      <SKU>cw56c1ddfe72660</SKU>
      <ImageType>PT1</ImageType>
      <ImageLocation>https://prod-plat-
almaenpena.geanetondemand.com//EPORTAL_IMGS/GENERAL/ALMAENPENA/IMG
2-ap56c1ddfe9d957/20830-2.jpg</ImageLocation>
    </ProductImage>
  </Message>
</AmazonEnvelope>
```

A.3 Mensaje de inventario (Stock)

```
<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>Inventory</MessageType>
  <Message>
    <MessageID>1</MessageID>
```



```

<Inventory>
  <SKU>cw56c1ddfe72660</SKU>
  <Available>true</Available>
</Inventory>
</Message>
</AmazonEnvelope>

```

A.4 Mensaje de relaciones

```

<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>Relationship</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <Relationship>
      <ParentSKU>cw56c1ddfe4b23f</ParentSKU>
      <Relation>
        <SKU>cw56c1ddfe72660</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>
        <SKU>cw56c1ddff61a67</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>
        <SKU>cw56c1de00382ef</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>
        <SKU>cw56c1de010accd</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>
        <SKU>cw56c1de01dd4da</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>
        <SKU>cw56c1de02d2f81</SKU>
        <Type>Variation</Type>
      </Relation>
      <Relation>

```

```
<SKU>cw56c1de03c9286</SKU>
<Type>Variation</Type>
</Relation>
<Relation>
  <SKU>cw56c1de04a9a93</SKU>
  <Type>Variation</Type>
</Relation>
</Relationship>
</Message>
</AmazonEnvelope>
```

B Ejemplo de comunicación de Pedidos con Amazon

B.1 Pedido realizado

```
<ListOrdersResponse>
  <ListOrdersResult>
    <LastUpdatedBefore>2016-06-13T08:28:08Z</LastUpdatedBefore>
    <Orders>
      <Order>
        <AmazonOrderId>XXXX</AmazonOrderId>
        <PurchaseDate>2016-06-13T07:56:17Z</PurchaseDate>
        <LastUpdateDate>2016-06-13T08:26:37Z</LastUpdateDate>
        <OrderStatus>Unshipped</OrderStatus>
        <FulfillmentChannel>MFN</FulfillmentChannel>
        <SalesChannel>Amazon.es</SalesChannel>
        <ShipServiceLevel>ES Std Domestic</ShipServiceLevel>
        <ShippingAddress>
          <Name>Javier Gutiérrez Navío</Name>
          <AddressLine1>Calle Alcalá, 119, 3D.</AddressLine1>
          <City>Madrid</City>
          <StateOrRegion>Madrid</StateOrRegion>
          <PostalCode>28009</PostalCode>
          <CountryCode>ES</CountryCode>
          <Phone>618350358</Phone>
        </ShippingAddress>
        <OrderTotal>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>3.00</Amount>
        </OrderTotal>
        <NumberOfItemsShipped>0</NumberOfItemsShipped>
        <NumberOfItemsUnshipped>1</NumberOfItemsUnshipped>
        <PaymentExecutionDetail></PaymentExecutionDetail>
        <PaymentMethod>Other</PaymentMethod>
        <MarketplaceId>A1RKKUPIHCS9HS</MarketplaceId>
        <BuyerEmail>XXXXXX@marketplace.amazon.es</BuyerEmail>
        <BuyerName>Javier Gutiérrez Navío</BuyerName>

        <ShipmentServiceLevelCategory>Standard</ShipmentServiceLevelCategory>
        <ShippedByAmazonTFM>false</ShippedByAmazonTFM>
        <OrderType>StandardOrder</OrderType>
        <EarliestShipDate>2016-06-13T22:00:00Z</EarliestShipDate>
        <LatestShipDate>2016-06-15T21:59:59Z</LatestShipDate>
        <EarliestDeliveryDate>2016-06-16T22:00:00Z</EarliestDeliveryDate>
        <LatestDeliveryDate>2016-06-21T21:59:59Z</LatestDeliveryDate>
        <IsPrime>false</IsPrime>
        <IsPremiumOrder>false</IsPremiumOrder>
      </Order>
    </Orders>
  </ListOrdersResult>
</ListOrdersResponse>
```

```

</ListOrdersResult>
<ResponseMetadata>
  <RequestId>db662937-c34b-45bb-8431-ea009bbc1ff2</RequestId>
</ResponseMetadata>
</ListOrdersResponse>

```

B.2 Productos de un pedido

```

<ListOrderItemsResponse>
  <ListOrderItemsResult>
    <AmazonOrderId>XXXXXX</AmazonOrderId>
    <OrderItems>
      <OrderItem>
        <ASIN>XXXXXXX</ASIN>
        <SellerSKU> XXXXXX </SellerSKU>
        <OrderItemId> XXXXXX </OrderItemId>
        <Title>V16002 - Prueba de pedidos</Title>
        <QuantityOrdered>1</QuantityOrdered>
        <QuantityShipped>0</QuantityShipped>
        <ItemPrice>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>1.00</Amount>
        </ItemPrice>
        <ShippingPrice>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>2.99</Amount>
        </ShippingPrice>
        <GiftWrapPrice>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>0.00</Amount>
        </GiftWrapPrice>
        <ItemTax>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>0.00</Amount>
        </ItemTax>
        <ShippingTax>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>0.00</Amount>
        </ShippingTax>
        <GiftWrapTax>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>0.00</Amount>
        </GiftWrapTax>
        <ShippingDiscount>
          <CurrencyCode>EUR</CurrencyCode>
          <Amount>0.00</Amount>
        </ShippingDiscount>
        <PromotionDiscount>
          <CurrencyCode>EUR</CurrencyCode>

```

```

        <Amount>0.00</Amount>
      </PromotionDiscount>
      <PromotionIds></PromotionIds>
      <ConditionId>New</ConditionId>
      <ConditionSubtypeId>New</ConditionSubtypeId>
    </OrderItem>
  </OrderItems>
</ListOrderItemsResult>
<ResponseMetadata>
  <RequestId>9fc3f301-cfe7-48d4-9e0d-40a75658a0a7</RequestId>
</ResponseMetadata>
</ListOrderItemsResponse>

```

B.3 Aceptación de un pedido

```

<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>OrderAcknowledgement</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <OrderAcknowledgement>
      <AmazonOrderID>XXXX</AmazonOrderID>
      <MerchantOrderID>XXXX</MerchantOrderID>
      <StatusCode>Success</StatusCode>
    </OrderAcknowledgement>
  </Message>
</AmazonEnvelope>

```

B.4 Rechazo de un pedido

```

<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>OrderAcknowledgement</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <OrderAcknowledgement>
      <AmazonOrderID>XXXX</AmazonOrderID>
      <StatusCode>Faillure</StatusCode>
    </OrderAcknowledgement>
  </Message>
</AmazonEnvelope>

```

```
</Message>
</AmazonEnvelope>
```

B.5 Cumplimentación de envío de un pedido

```
<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>OrderFulfillment</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <OrderFulfillment>
      <AmazonOrderID>XXXXXX</AmazonOrderID>
      <FulfillmentDate>2016-06-14T01:16:01Z</FulfillmentDate>
      <FulfillmentData>
        <CarrierName>Hermes Logistik Gruppe</CarrierName>
      </FulfillmentData>
      <Item>
        <AmazonOrderItemCode> XXXXX </AmazonOrderItemCode>
        <MerchantFulfillmentItemID>1</MerchantFulfillmentItemID>
      </Item>
    </OrderFulfillment>
  </Message>
</AmazonEnvelope>
```

B.6 Cancelación de un producto dentro de un pedido

```
<?xml version="1.0" encoding="utf-8"?>
<AmazonEnvelope>
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>XXXX</MerchantIdentifier>
  </Header>
  <MessageType>OrderFulfillment</MessageType>
  <Message>
    <MessageID>1</MessageID>
    <OrderAdjustment>
      <AmazonOrderID>XXXX</AmazonOrderID>
      <ActionType>Cancel</ActionType>
      <AdjustedItem>
        <AmazonOrderItemCode>XXXX</AmazonOrderItemCode>
        <AdjustmentReason>NoInventory</AdjustmentReason>
        <QuantityCancelled>2</QuantityCancelled>
      </AdjustedItem>
    </OrderAdjustment>
  </Message>
</AmazonEnvelope>
```

```
</OrderAdjustment>  
</Message>  
</AmazonEnvelope>
```

C Listado de links para los XSDs

Aquí podemos ver un listado con los principales XSD que se han usado para la validación de los mensajes XML.

- **Ajuste de pedido:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/OrderAdjustment.xsd
- **Amazon (más recursos):** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Amazon.xsd
- **Cabecera de paquetes:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/amzn-headers.xsd
- **Centros de envío:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/FulfillmentCenter.xsd
- **Confirmación de pedido:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/OrderAcknowledgement.xsd
- **Envelopes:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/amzn-envelopes.xsd
- **Envío de pedido:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/OrderFulfillment.xsd
- **Estado de procesamiento de pedido:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/OrderReport.xsd
- **Imagenes de un producto:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/ProductImages.xsd
- **Inventario:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Inventory.xsd
- **Precio:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Price.xsd
- **Producto:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Product.xsd
- **Recursos básicos:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/amzn-base.xsd
- **Relaciones:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Relationship.xsd
- **Zapatos:** https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/xsd/release_1_9/Shoes.xsd

D Ampliación del diagrama de clases para los Productos

